



Softwarebeschreibung Qt Visualisierung für Linux

Version: 2023/26

Teile-Nr.: 203744

"Original Dokumentation"

AMK*motion*

MEMBER OF THE ARBURG FAMILY

Impressum

Name: PDK_203744_Software_Qt-Visualisierung

Version:

Version: 2023/26

Änderung

Kurzzeichen

AMKmotion Design

LeS

Bisherige Version: 2017/38

Produktstand:

Produkt

Firmware Version (Teile-Nr.)

PC Software

AMK Qt Visualisierung Version 2.00 (O865)

Schutzvermerk:

© AMKmotion GmbH + Co KG

Weitergabe sowie Vervielfältigung dieser Unterlagen, Verwertung und Mitteilung ihres Inhalts wird nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zum Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

Vorbehalt:

Änderungen im Inhalt der Dokumentation und Liefermöglichkeit der Produkte sind vorbehalten.

Herausgeber:

AMKmotion GmbH + Co KG

Gaußstraße 37-39

73230 Kirchheim unter Teck

Germany

Phone +49 7021 50 05-0

Fax +49 7021 50 05-176

E-Mail info@amk-motion.com

Registergericht: AG Stuttgart, HRA 230681, Kirchheim unter Teck,

Ust.-Id.-Nr.: DE 145 912 804

Komplementär: AMKmotion Verwaltungsgesellschaft mbH, HRB 774646

Service:

Phone +49 7021 50 05-190, Fax -193

Zur schnellen und zuverlässigen Behebung der Störung tragen Sie bei, wenn Sie unseren Service informieren über:

- die Typenschildangaben der Geräte
- die Softwareversion
- die Gerätekonstellation und die Applikation
- die Art der Störung, vermutete Ausfallursache
- die Diagnosemeldungen (Fehlernummern)

E-Mail service@amk-motion.com

Internetadresse:

www.amk-motion.com

Inhalt

1 Darstellungskonventionen	5
2 Produktbezeichnung und Bestelldaten	6
2.1 Unterschiede CODESYS V2 / CODESYS V3	6
2.2 Hinweis zur GPL (General Public License)	6
3 Installation	7
3.1 Installationsvoraussetzungen AMK Qt Visualisierung	7
3.1.1 Hinweis zur AMK Steuerung	7
3.2 Installation AMK Qt Visualisierung	7
3.3 Installation 'Oracle VirtualBox'	11
3.4 Installation AMK Qt Visualisierung beenden	16
3.5 Installation 'Qt AddOn für CODESYS V3'	17
3.6 Deinstallation einer älteren Version	19
4 Qt Creator	23
4.1 Einführung	23
4.2 Erweiterung AMK Symbol Config	23
4.2.1 Starten der Erweiterung 'AMK Symbol Config'	23
4.2.2 Übersicht der Erweiterung 'AMK Symbol Config'	23
4.2.3 Funktionen der Erweiterung 'AMK Symbol Config'	24
4.2.3.1 Variablen konfigurieren	24
4.2.3.2 Variablen entfernen	24
4.2.3.3 Task konfigurieren	25
4.2.3.4 Task zuweisen	26
4.2.3.5 Konfiguration übernehmen und Code erzeugen	27
4.2.3.6 Konfiguration entfernen	27
4.2.3.7 Konfigurationsdateien	27
4.2.3.8 Funktion einer zyklischen Variablen aufrufen	27
4.3 Code Generierung	28
4.3.1 Visu Variablennamen	28
4.3.2 Code zur Verwendung von Tasks	30
5 First Steps CODESYS V2	32
5.1 Beschreibung First Steps	32
5.2 Grundeinstellungen Qt Creator	33
5.3 Grundeinstellungen Oracle VirtualBox	36
5.4 AIPEX PRO und CODESYS	37
5.4.1 Grundeinstellungen für AIPEX PRO	37
5.4.2 AIPEX PRO Projekt erzeugen	40
5.4.3 PLC Projekt erzeugen	41
5.4.4 Grundeinstellungen CODESYS (Symboldatei)	42
5.4.5 Testvariable anlegen	44
5.4.6 Symboldatei konfigurieren	44
5.4.7 Symboldatei erzeugen	47
5.5 Qt Creator	47
5.5.1 Wizard 'Neues Qt Projekt' erzeugen	47
5.5.2 Datenaustausch zwischen PLC und Qt	51
5.5.3 Qt Editor	55
5.5.4 Qt Visualisierung übersetzen	60
5.6 CODESYS fertigstellen	61
5.6.1 Qt Visualisierung unter CODESYS V2 integrieren	61
5.6.1.1 CODESYS V2 Projekt Fertigstellen und Testen	62
5.7 Weitere Beispiele	65
5.7.1 Button (tastend)	65

5.7.2 Button (toggelnd)	67
5.7.3 Istwerte anzeigen (Line Edit)	68
5.7.4 Istwerte anzeigen (ManoMeter)	70
5.7.5 Sollwerte vorgeben (über Numpad)	71
5.7.6 Statussignale (LED)	73
5.8 Tipps	75
5.8.1 Anwenderspezifischer Ordner	75
5.8.2 Umstellen eines vorhandenen Projektes auf ein anderes Zielsystem	79
5.8.3 Unterscheidung der Steuerungstypen	81
6 First Steps CODESYS V3	82
6.1 Symbolkonfiguration hinzufügen	82
6.2 Erweiterung im Qt Quellcode	85
6.3 Qt Visualisierung integrieren	86
6.3.1 Projekt übertragen	91
6.3.2 Pfadname absolut / relativ	93
7 Programmierung QtPlcHandler	95
7.1 Konfiguration und Verbindung	95
7.2 Status des QtPlcHandlers	96
7.3 Symbol-Informationen	96
7.4 Zyklisches Update von Variablen	97
7.5 SPS Programm steuern	98
7.6 Manuelles Lesen von Variablen	99
7.7 Manuelles Schreiben von Variablen	100
7.8 Datei Operationen	101
7.9 SPS Projektinformationen	103
7.10 QtPlcHandler Signale	104
8 Anhang	105
8.1 GNU General Public License (GPL V2)	105
8.1.1 Preamble	105
8.1.2 Terms and conditions for copying, distribution and modification	105
8.2 GNU General Public License (GPL V3)	108
8.2.1 Preamble	108
8.2.2 Terms and conditions	108
8.3 GNU Lesser General Public License (LGPL V2.1)	114
8.3.1 Preamble	114
8.3.2 Terms and conditions for copying, distribution and modification	115
Ihre Meinung zählt!	119

1 Darstellungskonventionen

Darstellung	Bedeutung
	Mit diesem Symbol wird auf Textstellen hingewiesen, die Ihre besondere Aufmerksamkeit verdienen.
	Parameternamen, z.B. "ID2 SERCOS Zykluszeit" Diagnosemeldung, z.B. "1234 Netzausfall"
0x	0x gefolgt von einer Hexadezimalzahl, z.B. 0x500A
'Namen' 'ID0815 Parameter text' '1234 Diagnosemeldung'	z.B.: Die Funktion 'PLC Programm löschen' aufrufen.
'fett'	Menüpunkte und Tasten in einer Software oder Bedieneinheit, z.B.: Klicken Sie den Taster 'OK' im Menü 'Optionen' , um die Funktion 'PLC Programm löschen' aufzurufen
>Eingabevariable<	Variablen, die in der Bedienoberfläche eingegeben werden.
<Variable>	Anwenderspezifische Variable z.B. boTest
	Das Symbol Hand zeigt in den Beispielen wo geklickt werden muss RMT: Rechte Maus Taste

2 Produktbezeichnung und Bestelldaten

Produktbezeichnung	Bestellnummer
AMK PC Software Qt Visualisierung	O865

Die AMK Bestellnummer O865 enthält die Setupdateien für die PC Software 'Qt Creator'. Die zudem benötigte Software 'Oracle VirtualBox' wird automatisch über das Internet heruntergeladen.

2.1 Unterschiede CODESYS V2 / CODESYS V3

Siehe 'First Steps CODESYS V3' auf Seite 82.

2.2 Hinweis zur GPL (General Public License)

Dieses Produkt verwendet freie Software, die unter GNU General Public License oder Lesser General Public License lizenziert ist:

- GPL Version 2
- GPL Version 3
- LGPL Version 2.1

Nachdem Sie das Produkt erworben haben, möchten Sie möglicherweise den Source Code der darin enthaltenen GPL Software bearbeiten , verändern oder weiterverteilen.

Auf Anfrage stellen wir Ihnen gerne die betroffenen Softwarequellen zur Verfügung.

Wichtige Anmerkung!

Wir übernehmen keinerlei Garantien und Gewährleistungen für die bereitgestellten Softwarequellen und geben dazu keinen Support.

Der vollständige Lizenztext liegt dem Produkt bei Auslieferung bei bzw. finden Sie in der Softwarebeschreibung Qt im Kapitel Anhang.

Weitere Informationen finden Sie bei www.gnu.org.

3 Installation

3.1 Installationsvoraussetzungen AMK Qt Visualisierung

PC System-Voraussetzungen:

- Windows XP, 7
- Prozessor mind. 1 GHz
- Festplatte Speicherplatz mind. 8 GB
- Ramspeicher mind. 1 GB
- Internetverbindung (um Oracle Virtual Box herunterzuladen)

3.1.1 Hinweis zur AMK Steuerung

Firmware AS_404_1120_203548



Die Firmware AS_404 (oder höher) muss komplett installiert werden. Das Updatefile AS_404_1120_203548_Update.zip ist nicht ausreichend.

In der Steuerung muss die Option 'Qt Visualisierung' aktiviert werden. Siehe Dokumentation PDK_202975_A4-A5_Produktbeschreibung Kapitel Inbetriebnahme A5D mit Tastatur.

3.2 Installation AMK Qt Visualisierung

Starten der Installation

Nach dem Einlegen der CD startet die Installation automatisch.

Ist der automatische Start deaktiviert, muss die Datei 'AMK Qt Visualisierung' - 'Setup.exe' aufgerufen werden.

Auswahl der Installationssprache



Sprache auswählen, mit 'OK' bestätigen

Start der Installation



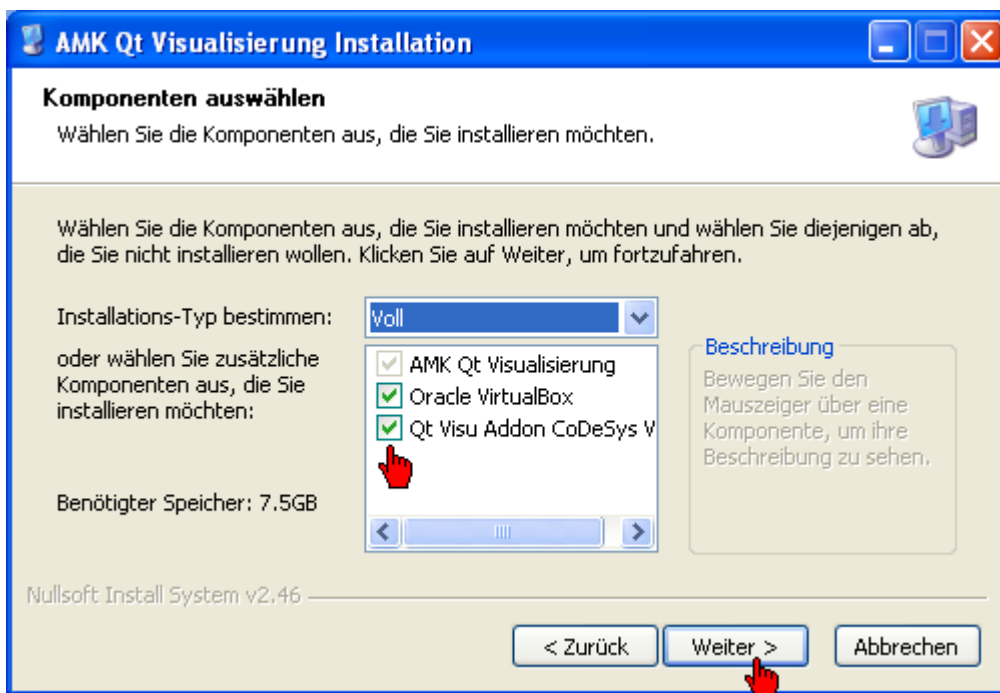
Button **'Weiter'** betätigen

Installationskomponenten

Ist die Oracle VirtualBox auf dem PC vorhanden, muss diese nicht erneut installiert werden. Die VirtualBox kann aber durch Anwahl 'Oracle VirtualBox' aktualisiert werden.

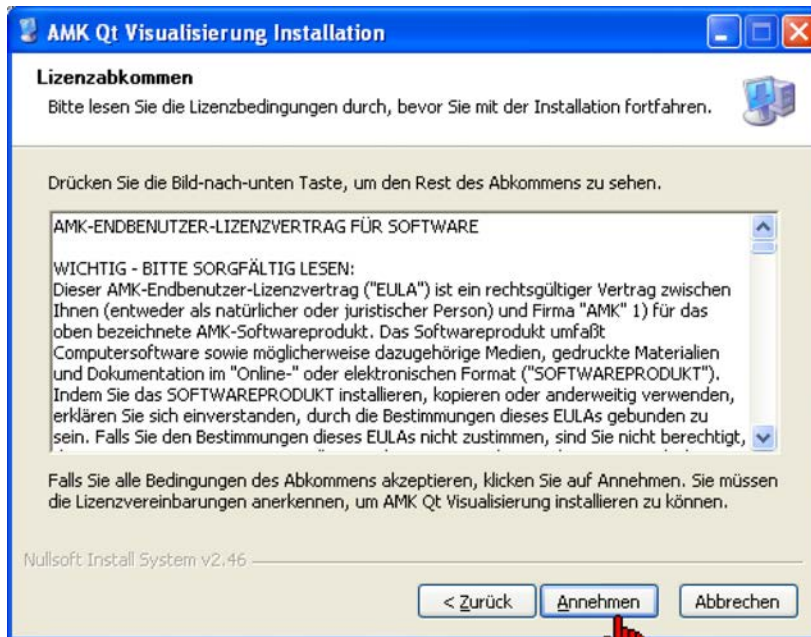


Wenn Sie CODESYS V3 verwenden muss das 'Qt Visu Addon CODESYS V3' installiert werden. [Siehe 'Installation 'Qt AddOn für CODESYS V3'' auf Seite 17.](#)



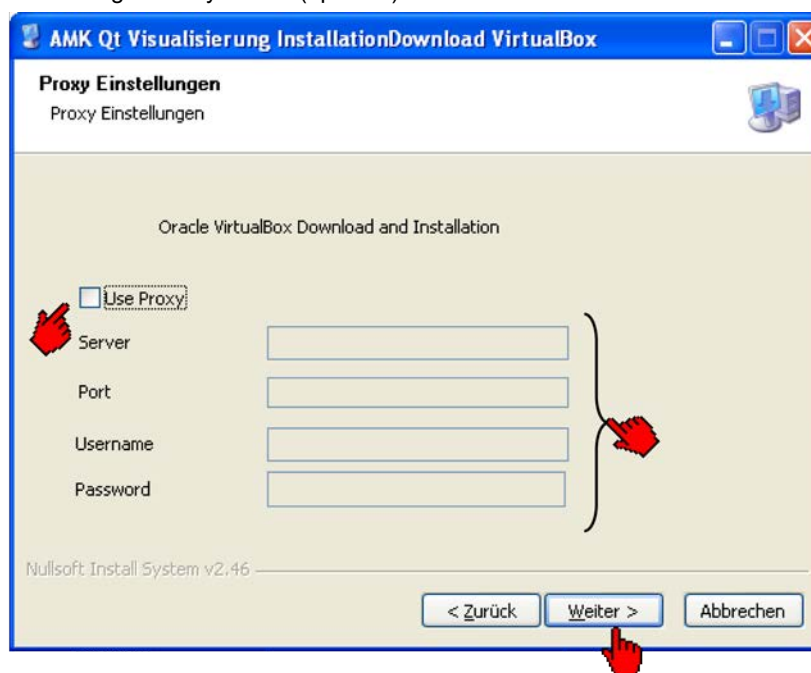
Button **'Weiter'** betätigen.

Lizenz akzeptieren



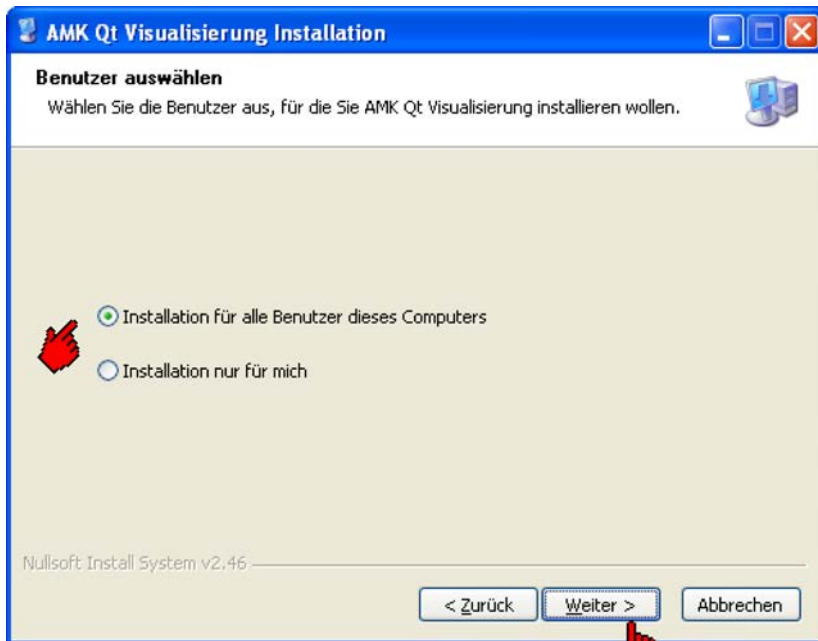
Button '**Annehmen**' betätigen

Einstellungen Proxy Server (optional)



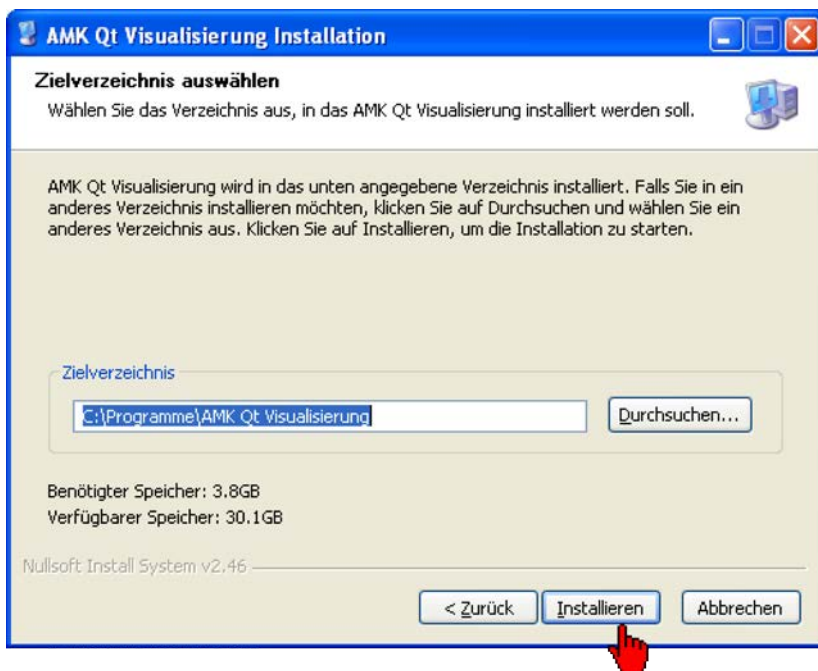
Im Normalfall werden die Einstellungen des Proxy Servers automatisch ermittelt, es muss nichts konfiguriert werden. In einzelnen Fällen ist es notwendig den Proxy Server manuell zu konfigurieren. Button '**Weiter**' betätigen.

Auswahl Benutzer



Button **'Weiter'** betätigen.

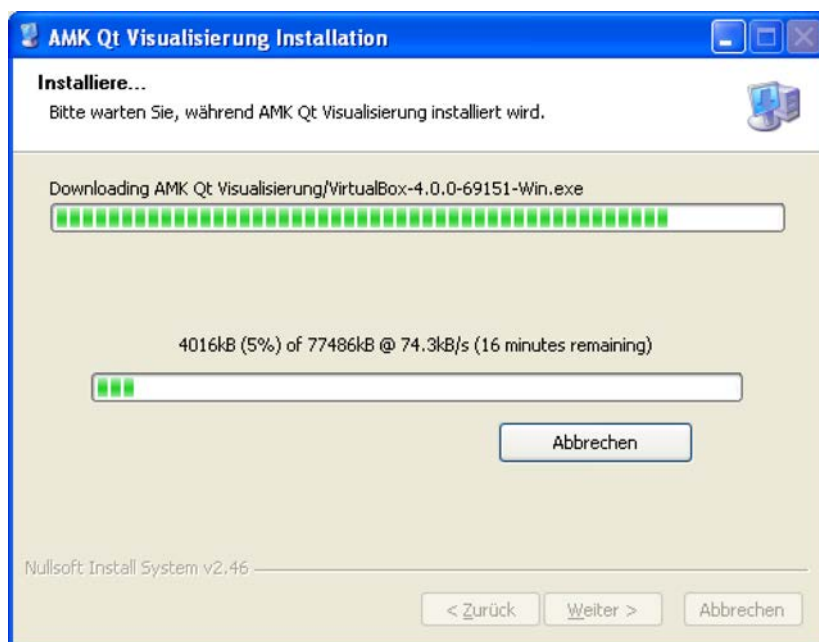
Installationsverzeichnis auswählen



Vorgegebenen Pfad übernehmen

Button **'Installieren'** betätigen.

Installationsfortschritt



3.3 Installation 'Oracle VirtualBox'

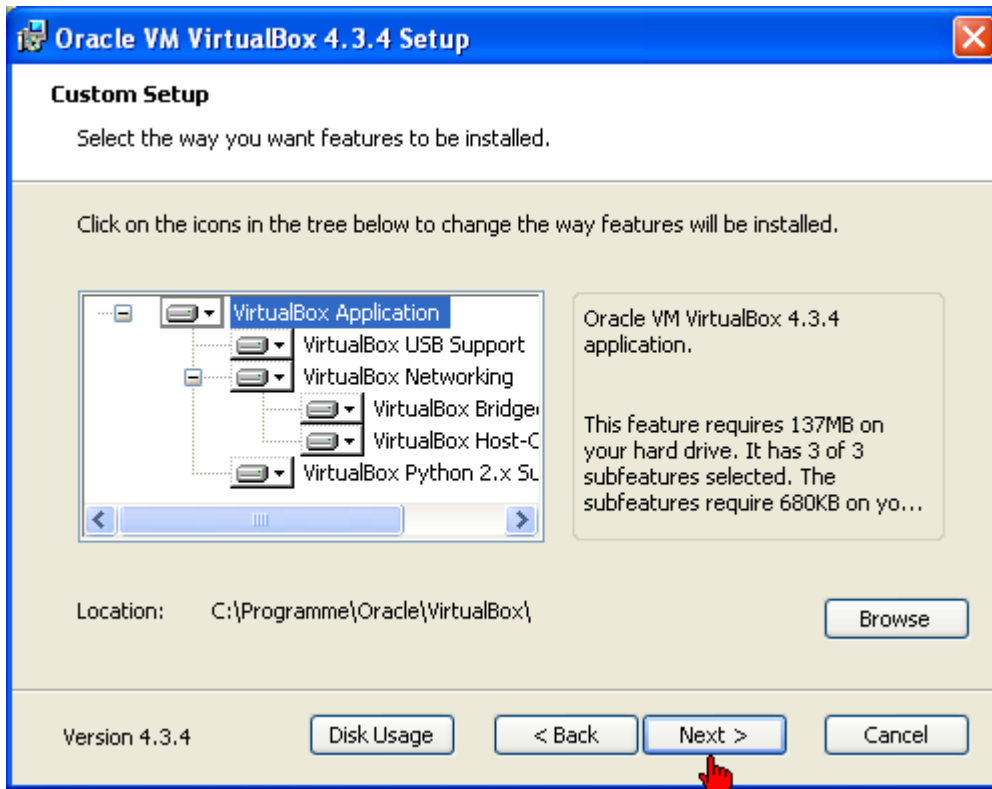
Start der Installation

Nach dem Download der 'Oracle VirtualBox' wird die Installation automatisch gestartet



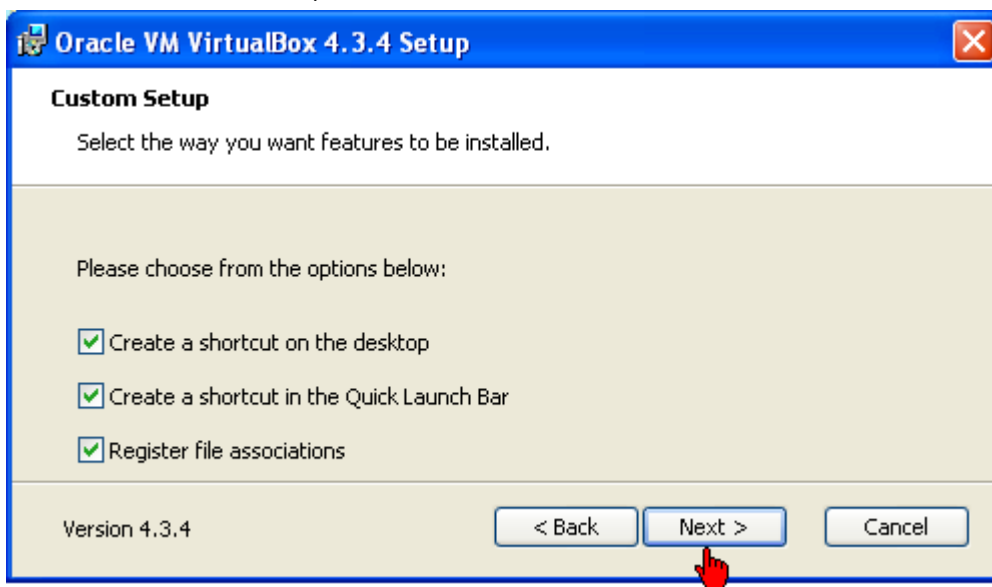
Button 'Next' betätigen

Installationskomponenten



Button 'Next' betätigen

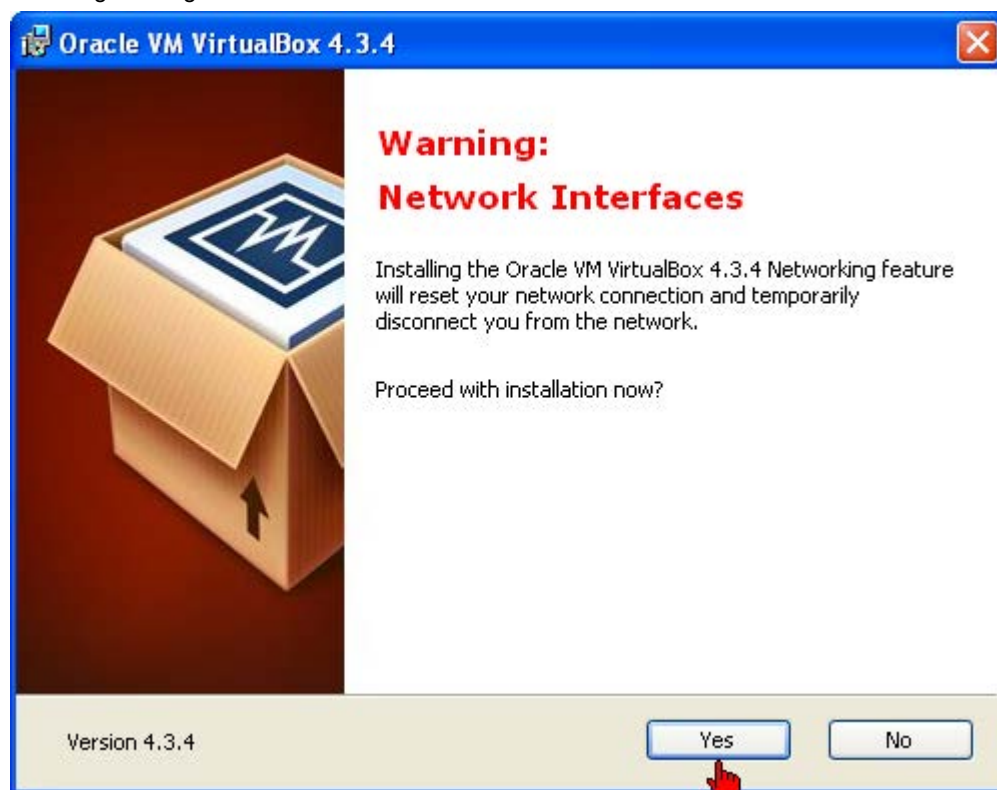
Schnellstartleiste und Desktop-Icon



Auswahl vornehmen

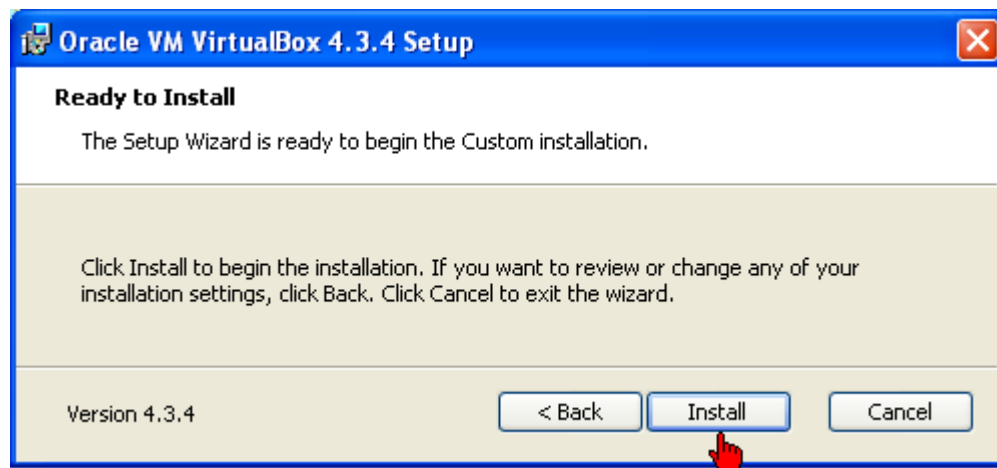
Button 'Next' betätigen

Warnung bestätigen



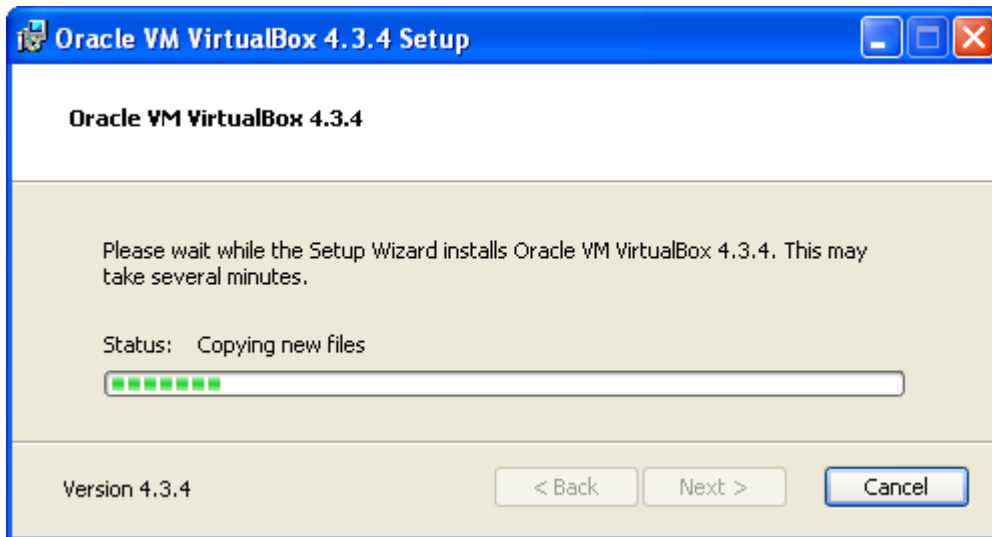
Während der Installation werden die Netzwerkverbindungen kurzzeitig unterbrochen.
Button **'Yes'** betätigen

Installation



Button **'Install'** betätigen

Installationsfortschritt

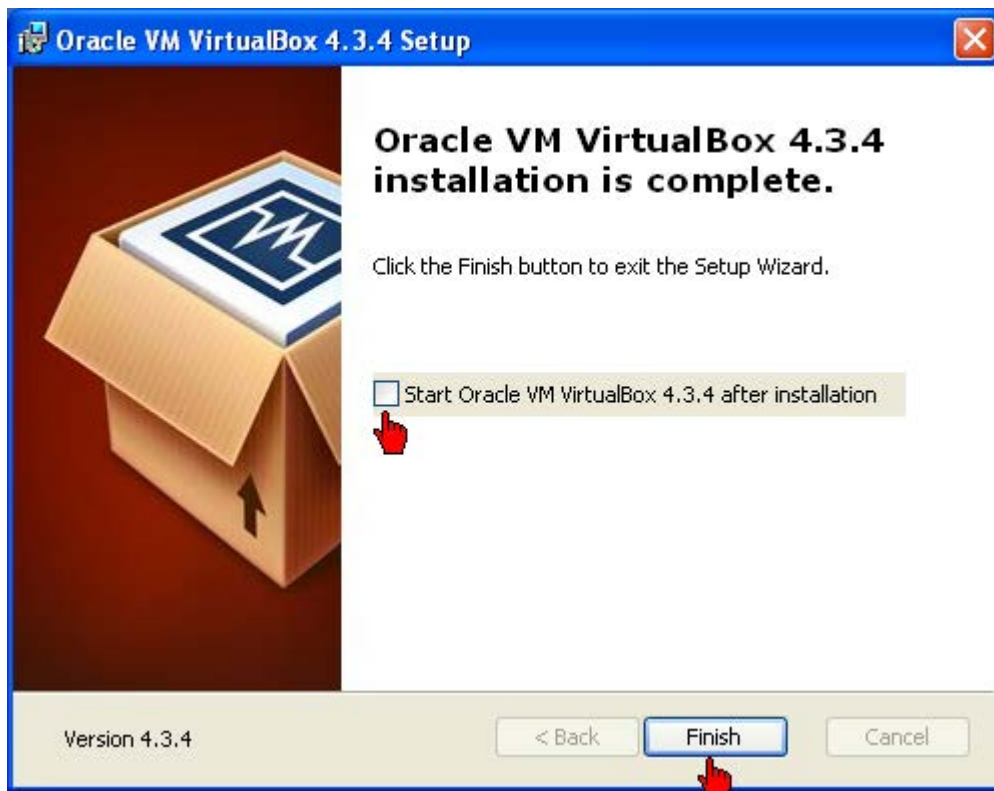


Treiber Installation (Windows XP)



Alle mit Button 'Installation fortsetzen' bestätigen

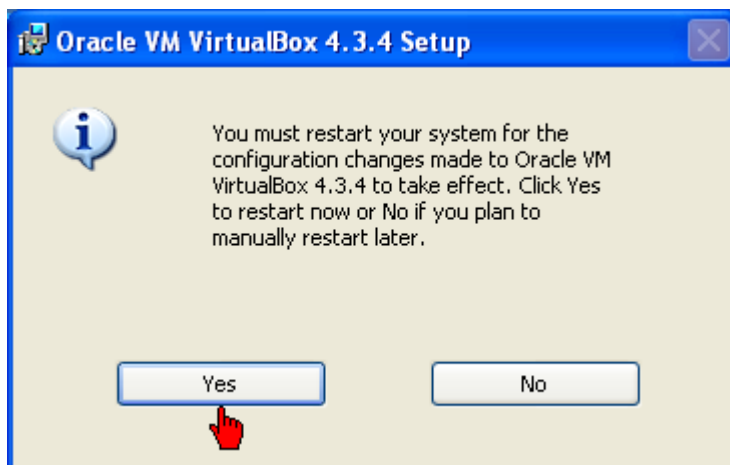
Installation beenden



Option 'Start Oracle VM VirtualBox 4.0.0 after installation' **abwählen**

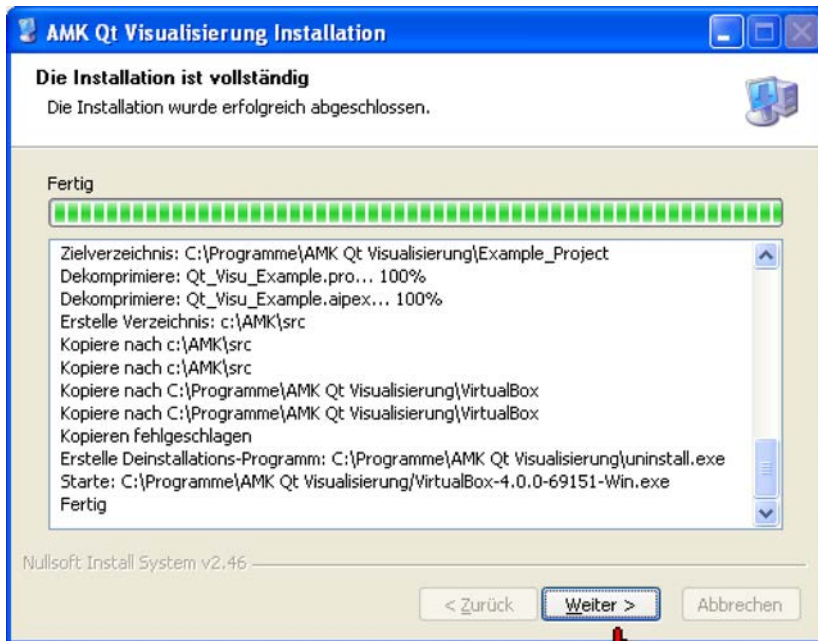
Button '**Finish**' betätigen

Neustart PC



3.4 Installation AMK Qt Visualisierung beenden

Installationsfortschritt



Button '**Weiter**' betätigen

Fertigstellen



Button '**Fertig stellen**' betätigen

3.5 Installation 'Qt AddOn für CODESYS V3'

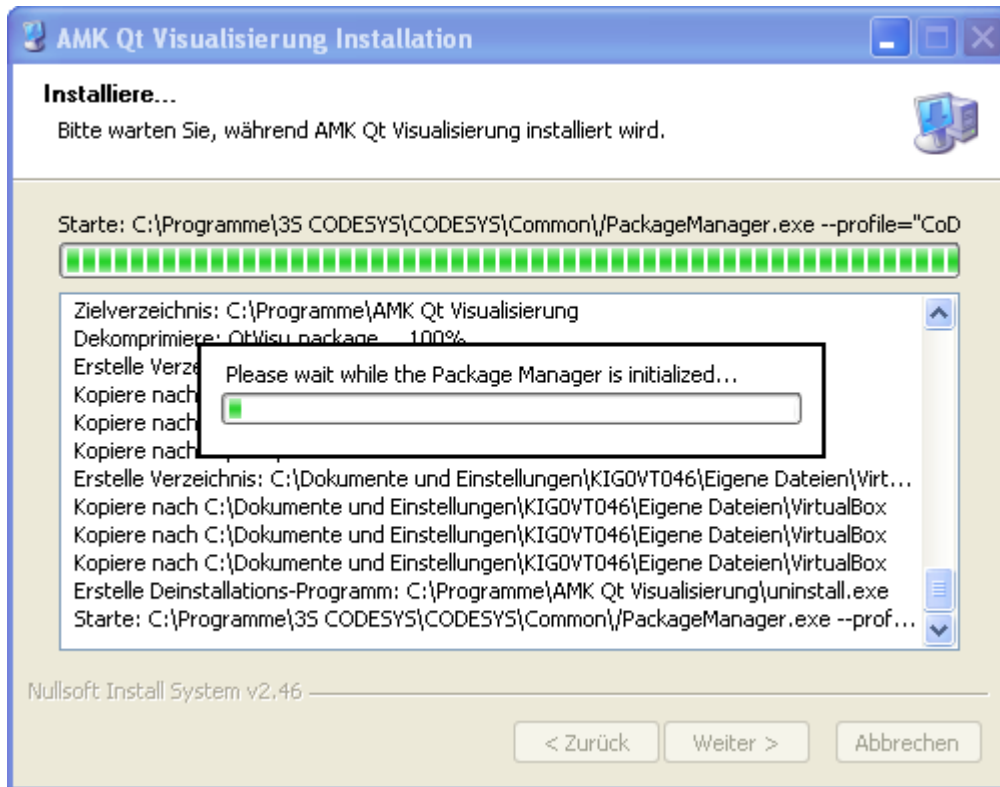


Wenn Sie CODESYS V3 verwenden müssen Sie das 'Qt AddOn für CODESYS V3' installieren.

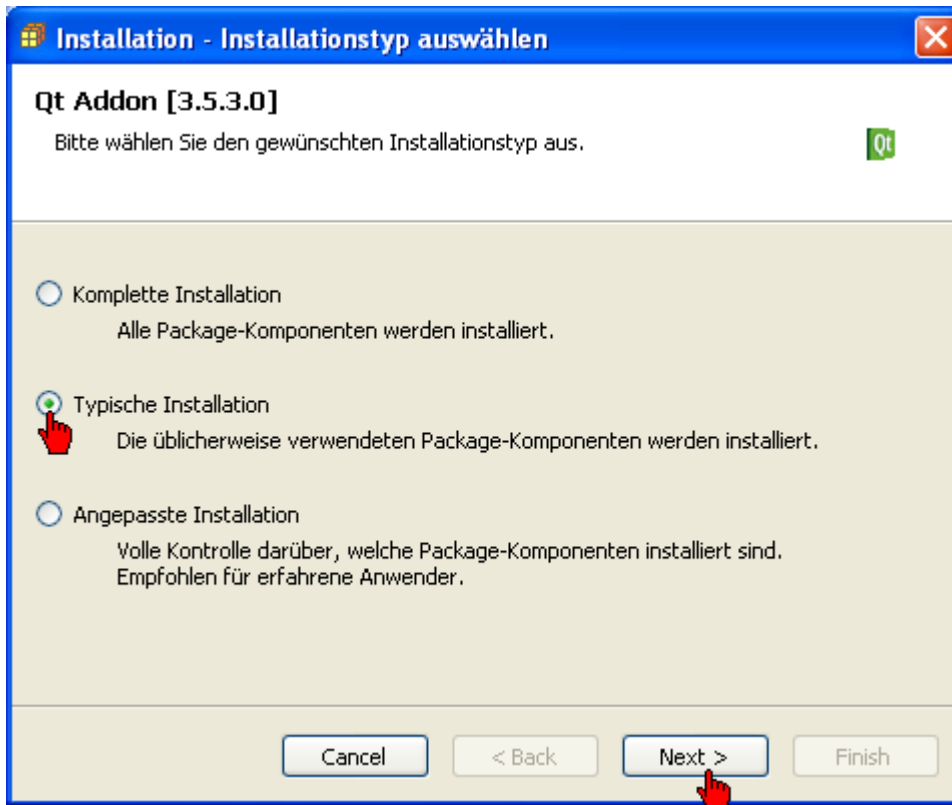
Nach einer Neuinstallation von CODESYS V3 auf einen höheren Stand, muss auch das 'Qt Add on' neu installiert werden.

Speicherort 'Qt Add On'

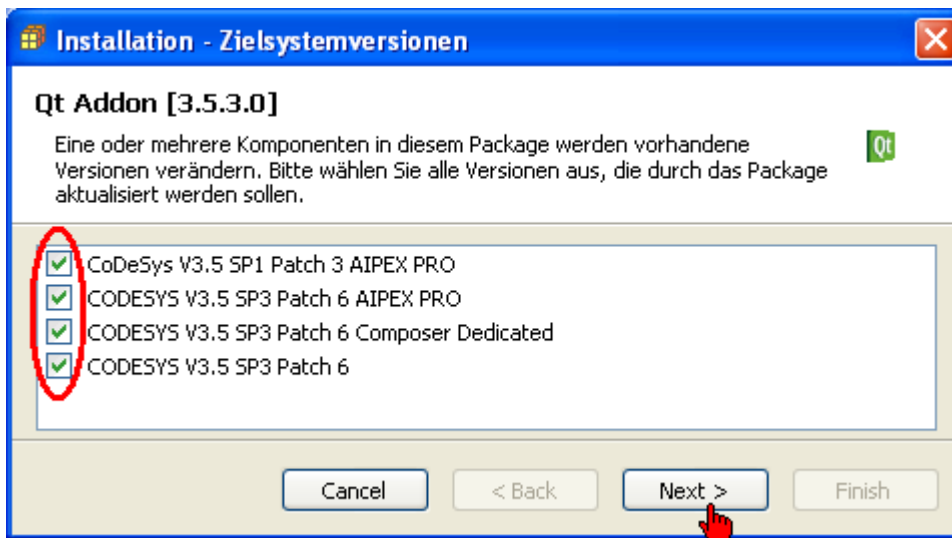
C:\Programme (x86)\AMK Qt Visualisierung



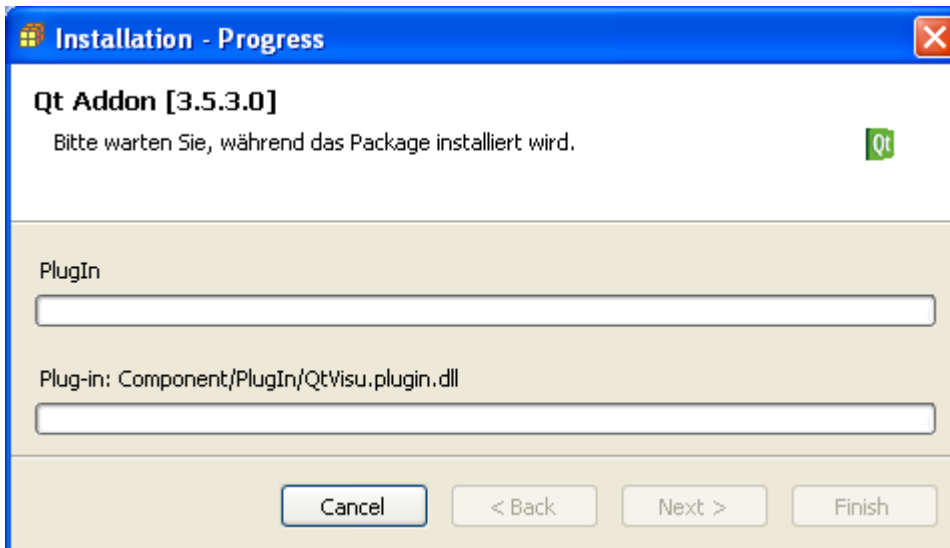
Installationstyp auswählen



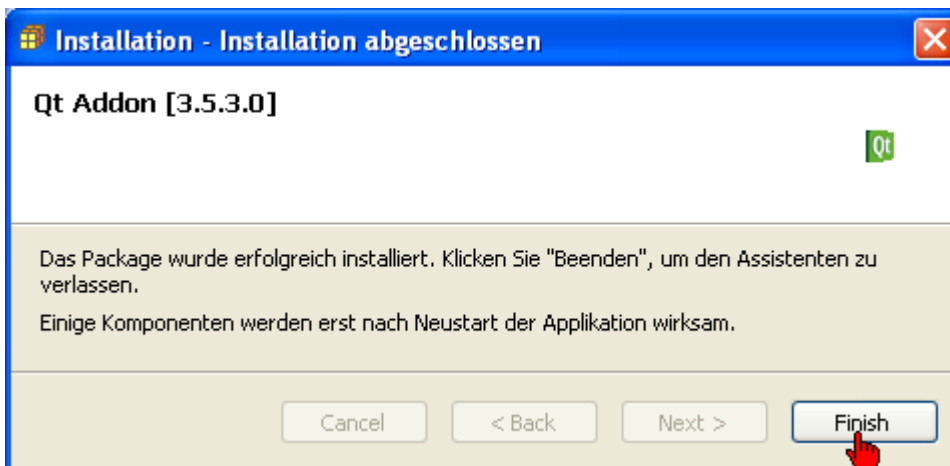
Installationskomponenten/Zielsystemversionen



Installationsfortschritt

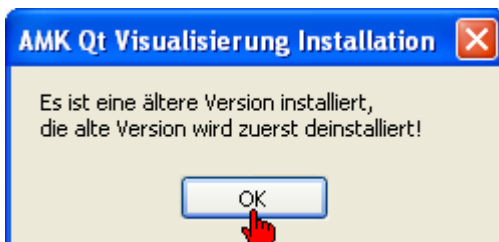


Installation beenden



3.6 Deinstallation einer älteren Version

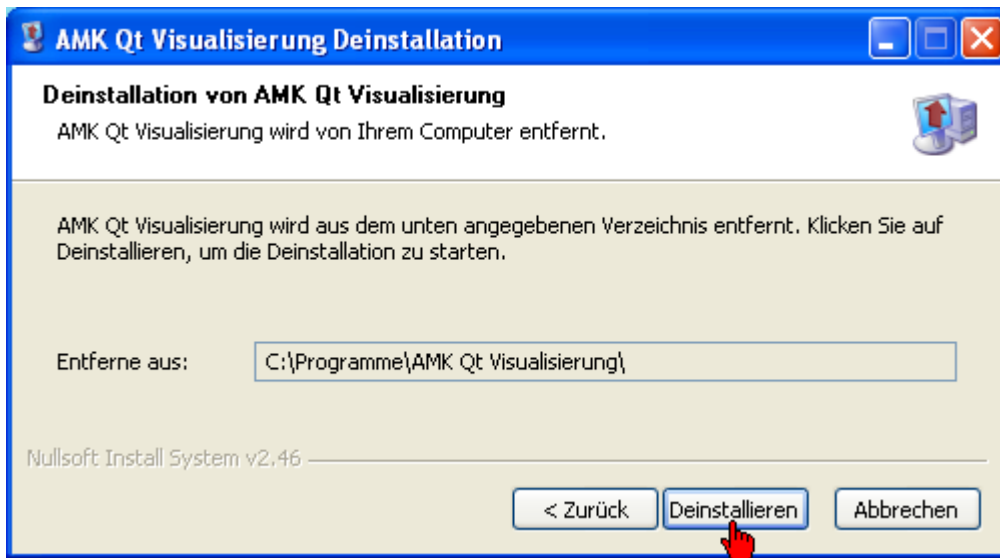
Wenn bereits eine ältere Version auf dem PC vorhanden ist, muss diese zuerst deinstalliert werden. Die Deinstallation wird vom 'AMK Qt Installer' automatisch durchgeführt.



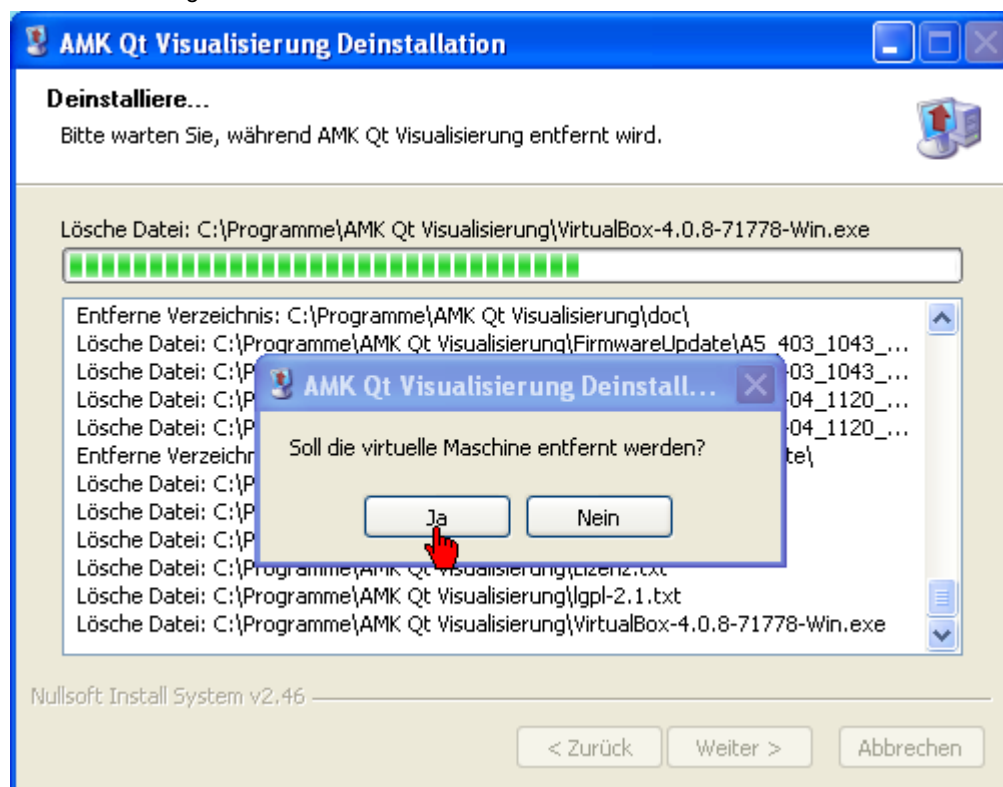
Starten Sie die Deinstallation



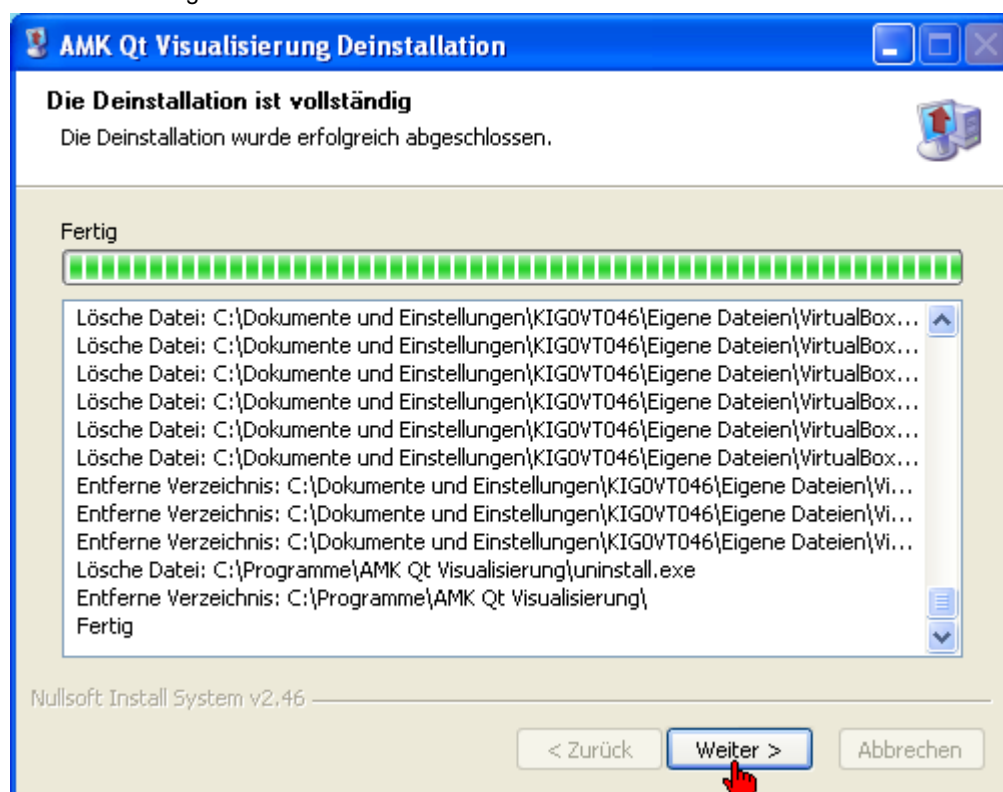
Deinstallationspfad



Fortschrittsanzeige der Deinstallation



Deinstallation abgeschlossen



Damit die Deinstallation vollständig abgeschlossen werden kann, muss der PC nach der Deinstallation der 'AMK Qt Visualisierung' neu gestartet werden.

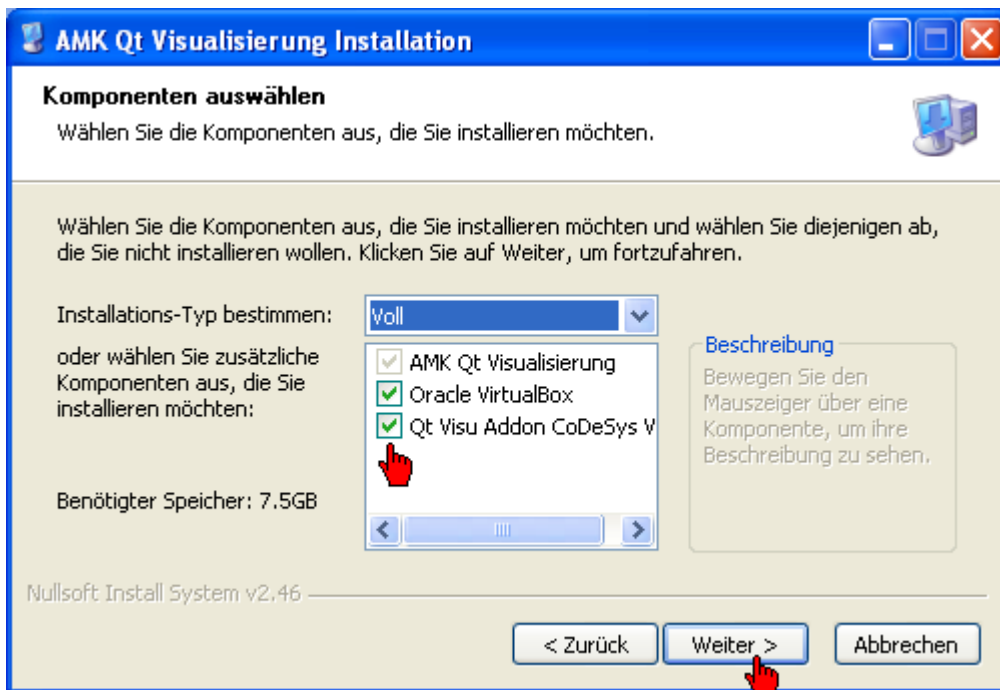


Installationskomponenten

Ist die Oracle VirtualBox auf dem PC vorhanden, muss diese nicht erneut installiert werden. Die VirtualBox kann aber durch Anwahl 'Oracle VirtualBox' aktualisiert werden.



Wenn Sie CODESYS V3 verwenden muss das 'Qt Visu Addon CODESYS V3' installiert werden. [Siehe 'Installation 'Qt AddOn für CODESYS V3'' auf Seite 17.](#)



Button **'Weiter'** betätigen.

Anschließend [siehe 'Installation AMK Qt Visualisierung' auf Seite 7](#)

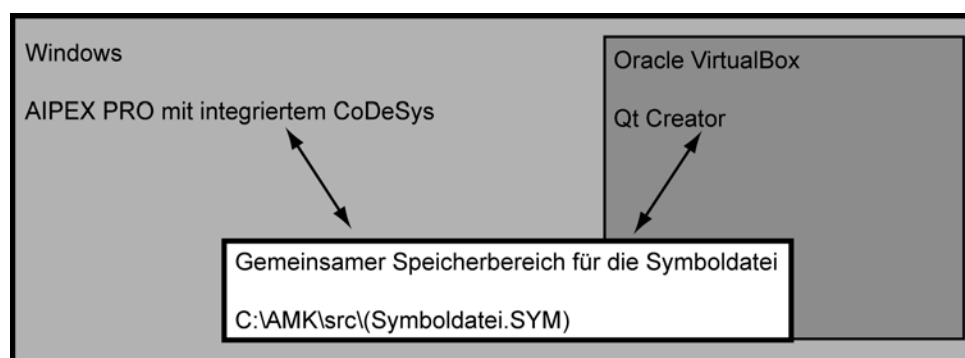
4 Qt Creator

4.1 Einführung

Der 'Qt Creator' ist eine integrierte Entwicklungsumgebung, mit der Sie Visualisierungsprogramme erstellen können. Die Programmierung erfolgt in C++.

Der 'Qt Creator' wird in der Oracle VirtualBox (im Dokument 'Virtuelle Maschine' genannt) gestartet. Das CODESYS PLC Programm und das Visualisierungsprogramm kommunizieren über den QtPlcHandler ([Siehe Programmierung QtPlcHandler auf Seite 95.](#)).

Der QtPlcHandler benötigt für den Datenaustausch eine Symboldatei. Die Symboldatei wird mit CODESYS erstellt und muss in einem für beide Programme zugänglichen Speicherbereich abgelegt werden. Für die einfachere Handhabung der Kommunikation bzw. der Symbole gibt es im 'Qt Creator' eine AMK Erweiterung (AMK Symbol Config).



4.2 Erweiterung AMK Symbol Config

4.2.1 Starten der Erweiterung 'AMK Symbol Config'

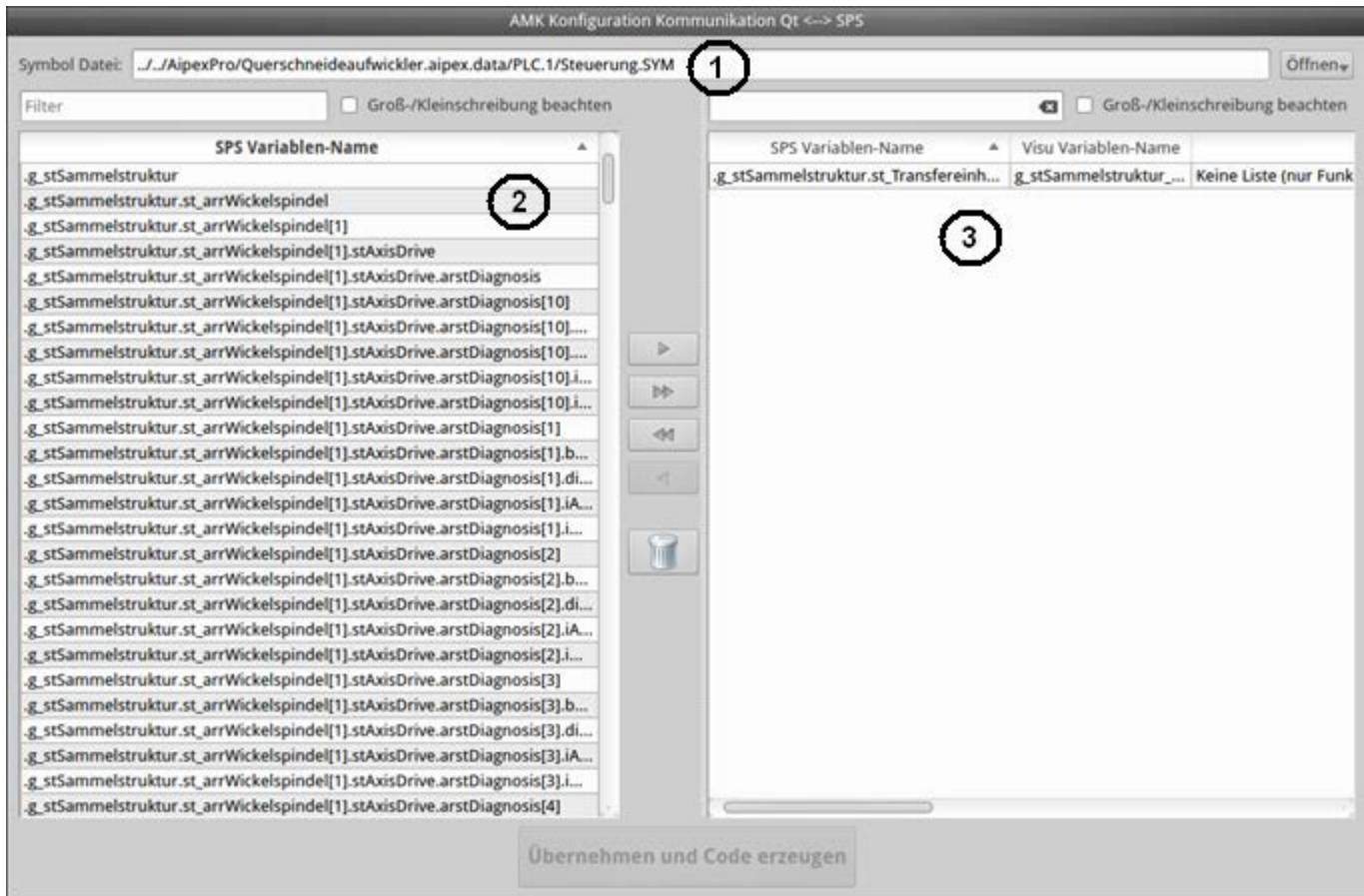
Die Erweiterung 'AMK Symbol Config' wird gestartet, indem auf der linken Seite der Modus 'AMK Symbol Config' angewählt wird alternativ durch die Tastenkombination 'STRG' + '7'.



4.2.2 Übersicht der Erweiterung 'AMK Symbol Config'

- Öffnen und aktualisieren der Symboldatei
- Übernahme der Variablen in die Qt Visualisierung
- Umbenennen der Variable (Alias Name)

- Taskverwaltung
- Automatische Code Erstellung



- 1 Ort und Name der Symboldatei
- 2 Liste mit allen Variablen, die in der Symbolkonfiguration angelegt wurden
- 3 Liste mit den Variablen, die im Visualisierungsprogramm verwendet werden sollen, ihren Alias-Namen und, ob sie zyklisch übertragen werden sollen

4.2.3 Funktionen der Erweiterung 'AMK Symbol Config'

4.2.3.1 Variablen konfigurieren

Es gibt verschiedene Möglichkeiten, Variablen anzulegen:

- Aus der Symbol-Liste ein oder mehrere Symbole auswählen und mit dem Button  die Variablen rechts in die Tabelle einfügen.
- Per Drag & Drop ein oder mehrere Symbole aus der Symbol-Liste nach rechts in die Variablen-Konfigurationstabelle ziehen.
- Mit dem Button  werden alle Variablen aus der Symbol-Liste übernommen.

4.2.3.2 Variablen entfernen

Es gibt verschiedene Möglichkeiten, Variablen zu entfernen:

- Aus der Variablen-Konfigurationstabelle eine oder mehrere Zeilen selektieren und mit dem Button  die ausgewählten Variablen entfernen.
- Per Drag & Drop eine oder mehrere Zeilen aus der Variablen-Konfigurationstabelle auswählen und nach links in die Symbol-Liste ziehen.
- Mit dem Button  werden alle Variablen aus der Variablen-Konfigurationstabelle entfernt.

4.2.3.3 Task konfigurieren

Für Ist- und Sollwerte sowie für boolsche Signale die zyklisch Übertragen werden, muss eine Task (Zeitebene) angelegt und eine Zykluszeit konfiguriert werden.

Sonst kann nur mit Read und Write (PLC_Value) zugegriffen werden.

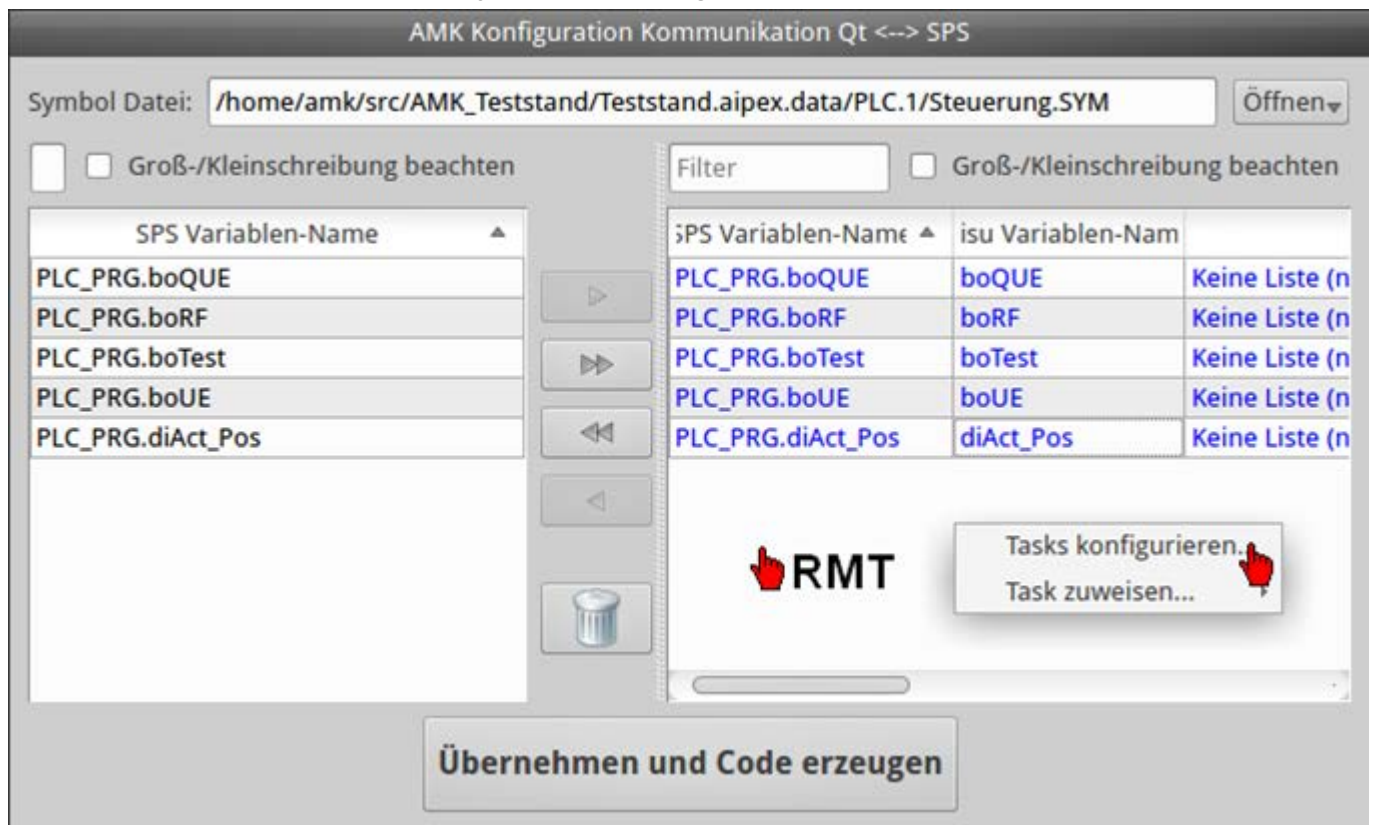


Die zyklisch konfigurierten Variablen werden nur bei Wertänderung übertragen.

Das bedeutet, auch wenn die Zykluszeit auf z.B. 100 ms eingestellt wurde, der Wert der Variablen sich aber nicht geändert hat, wird die Variablen auch nicht übertragen.

Beispiel: Neue Task konfigurieren

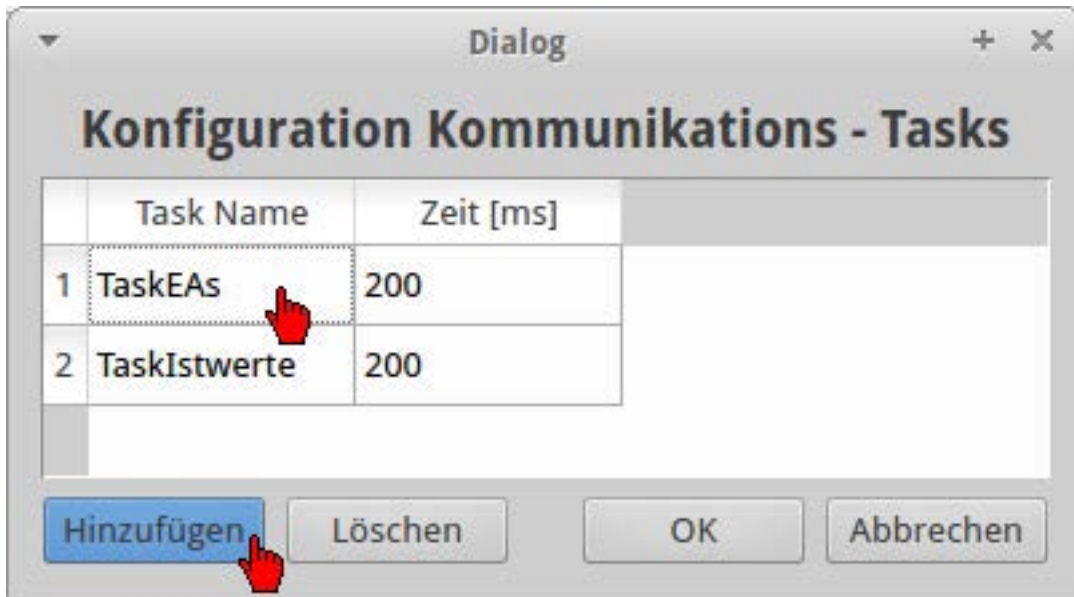
Klicken Sie in die Spalte **'Task'** um das Dialogfeld mit **'Task konfigurieren'** zu öffnen.



Klicken Sie auf den Taster **'Hinzufügen'** um eine neue Task anzulegen.
 Sie können jeder neuen Task einen spezifischen Namen vergeben.



Der Task Name darf nicht mit einer Zahl beginnen.

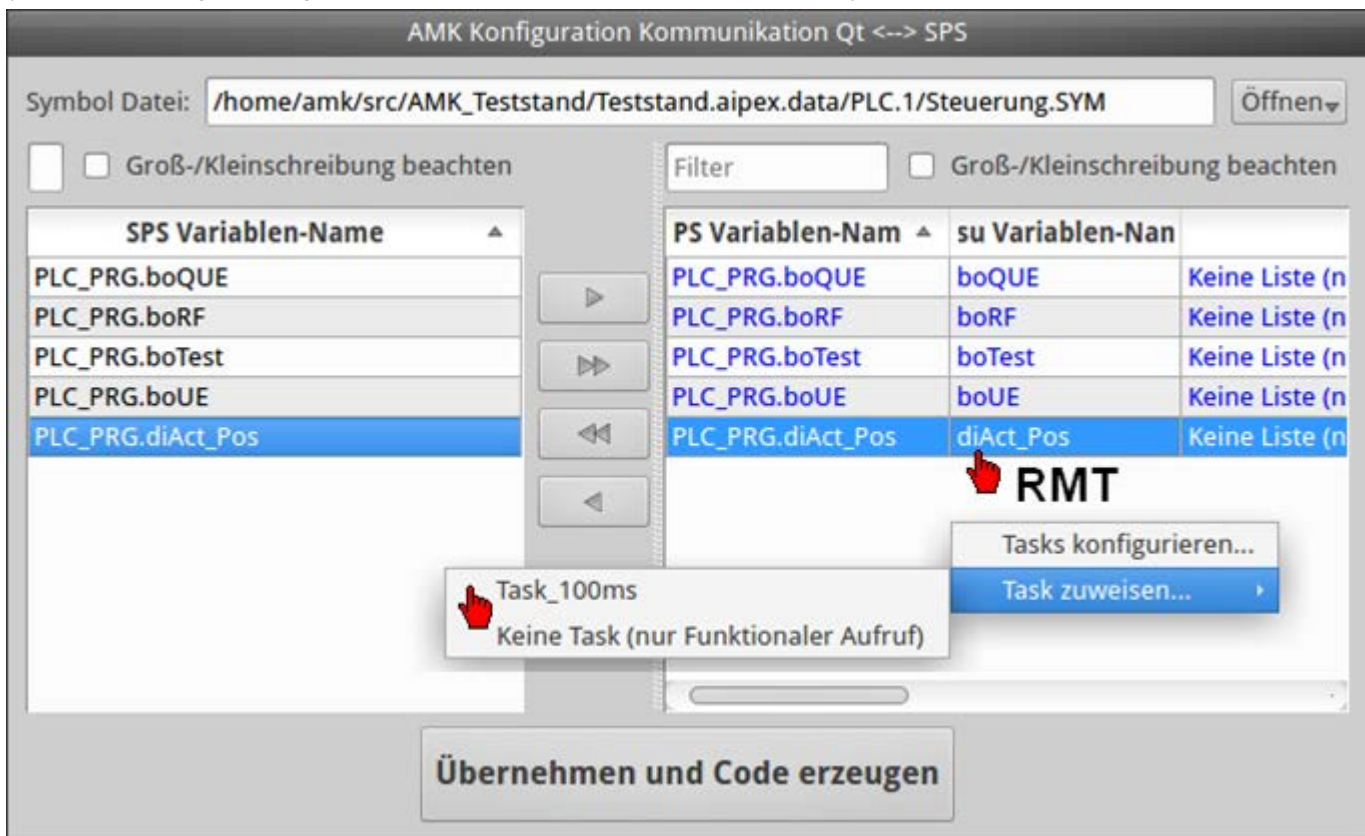


4.2.3.4 Task zuweisen

Klicken Sie mit der 'RMT' auf die Zeile, deren Variable Sie eine Task zuweisen möchten.

Anschließend können Sie über das Dialogfeld eine vorhandene Task zuweisen.

(Sie können auch gleichzeitig mehrere Zeilen markieren und eine Task zuweisen)



4.2.3.5 Konfiguration übernehmen und Code erzeugen

Übernehmen und Code erzeugen

Mit Betätigen des Buttons  wird die Konfiguration gespeichert und der notwendige Code erzeugt.

4.2.3.6 Konfiguration entfernen



Mit Betätigen des Buttons werden

- die Konfiguration entfernt
- die Symbol-Liste gelöscht
- die Variablen-Konfigurationstabelle gelöscht
- der Symbol Dateiname / Pfad gelöscht
- alle angelegten Tasks werden gelöscht

4.2.3.7 Konfigurationsdateien

Beim Erstellen der Konfiguration werden im Ordner des Projektes drei Dateien angelegt:

- In der Datei 'Projektname.ini' wird der Symbolname und Pfad gespeichert.
- In der Datei 'Projektname.dat' werden die Variablennamen und die Symbolnamen gespeichert.
- In der Datei 'Projektname.dac' wird die Task - Konfiguration gespeichert.



Die Konfigurationsdateien dürfen nicht manuell gelöscht werden, da sonst die Konfiguration und der Variablenbezug im Code verloren gehen.

4.2.3.8 Funktion einer zyklischen Variablen aufrufen



Quellcode muss vorher erzeugt werden.

Aus dem Kontextmenü der Variablen-Konfigurationstabelle kann die Option 'Gehe zu <Variablenname>' ausgewählt werden. Damit wird im Quellcode zu der Methode gesprungen, die aufgerufen wird, wenn der Wert der Variable sich ändert.

Klicken Sie mit der 'RMT' auf die Zeile, deren automatisch generierte Funktion Sie aufrufen möchten.
Klicken Sie anschließend auf 'Gehe zu '<Variablenname>'.

PS Variablen-Nam	su Variablen-Nam	Task
PLC_PRG.boQUE	boQUE	Keine Liste (nur Funktionaler Aufruf)
PLC_PRG.boRF	boRF	Keine Liste (nur Funktionaler Aufruf)
PLC_PRG.boTest	boTest	Keine Liste (nur Funktionaler Aufruf)
PLC_PRG.boUE	boUE	Keine Liste (nur Funktionaler Aufruf)
PLC_PRG.diAct_Pos	diAct_Pos	Task 100ms

Tasks konfigurieren...

Task zuweisen...

Gehe zu "diAct_Pos"

RMT

```

136
137 void MainWindow::on_diAct_Pos changed(const QVariant &value)
138 {
139     //Add your code below
140
141 }
  
```

4.3 Code Generierung

4.3.1 Visu Variablennamen

Die Visu Variablennamen aus der 'AMK Symbol Config' werden in die Headerdatei 'symbolConfig.h' geschrieben. Diese Datei wird automatisch in der Datei 'mainwindow.cpp' inkludiert, sodass die Variablen in der Hauptklasse zur Verfügung stehen.

Mit den Visu Variablennamen kann jetzt auf die Variablen im PLC Programm zugegriffen werden, z.B. durch die Verwendung der Methode 'readPlcValue(const QString &variable, IEC_BOOL &value);'. [Siehe Programmierung QtPlcHandler auf Seite 95.](#)



In der Datei 'symbolConfig' dürfen keine Änderungen vorgenommen werden, da diese Datei automatisch generiert wird und somit manuelle Änderungen verloren gehen.

Beispiel

```

symbolConfig.h*
diAct_Pos: const QString

1 ///////////////////////////////////////////////////////////////////
2 //This is a auto generated file.
3 //Do not edit this file, it will overwritten.
4 ///////////////////////////////////////////////////////////////////
5 #ifndef SYMBOLCONFIG_H
6 #define SYMBOLCONFIG_H
7 #include <QtCore/QString>
8 #include <QtCore/QStringList>
9
10 QStringList Task_100ms;
11 const QString boRF("PLC_PRG.boRF");
12 const QString boTest("PLC_PRG.boTest");
13 const QString boUE("PLC_PRG.boUE");
14 const QString boQUE("PLC_PRG.boQUE");
15 const QString diAct_Pos("PLC_PRG.diAct_Pos");
16
17 #endif // SYMBOLCONFIG_H
18

```

```

mainwindow.cpp*

1 #include "mainwindow.h"
2 #include "ui_mainwindow.h"
3 #include "symbolConfig.h"
4 #include <QtGui/QMessageBox>
5

```

Verwendung einer Variablen in 'mainwindow.cpp'

```

if (name==boQUE) {
    on_boQUE_changed(value);
}

```


Definition:

```

119
120 void MainWindow::on_PLG_PRG_diAct_Pos_changed(const QVariant &value)
121 {
122     //Add your code below
123 }
124

```

Diese Methoden werden automatisch aufgerufen, wenn der Wert der Variable sich geändert hat. Dafür wird eine Methode / ein Slot erzeugt, der die richtige Methode aufruft.

```

148
149 //do not edit this method, it will overwritten automatically
150 void MainWindow::cyclicData(const QList<PLC_VARIABLE> &data, const QList<int> &index)
151 {
152     for (int i=0; i<index.size(); ++i){
153         QString name=data.at(index.at(i)).name;
154         QVariant value=data.at(index.at(i)).value;
155         if (name==PLC_PRG_boQUE){
156             on_PLG_PRG_boQUE_changed(value);
157         }
158         if (name==PLC_PRG_boSBM){
159             on_PLG_PRG_boSBM_changed(value);
160         }
161         if (name==PLC_PRG_diAct_Pos){
162             on_PLG_PRG_diAct_Pos_changed(value);
163         }
164     }
165 }

```



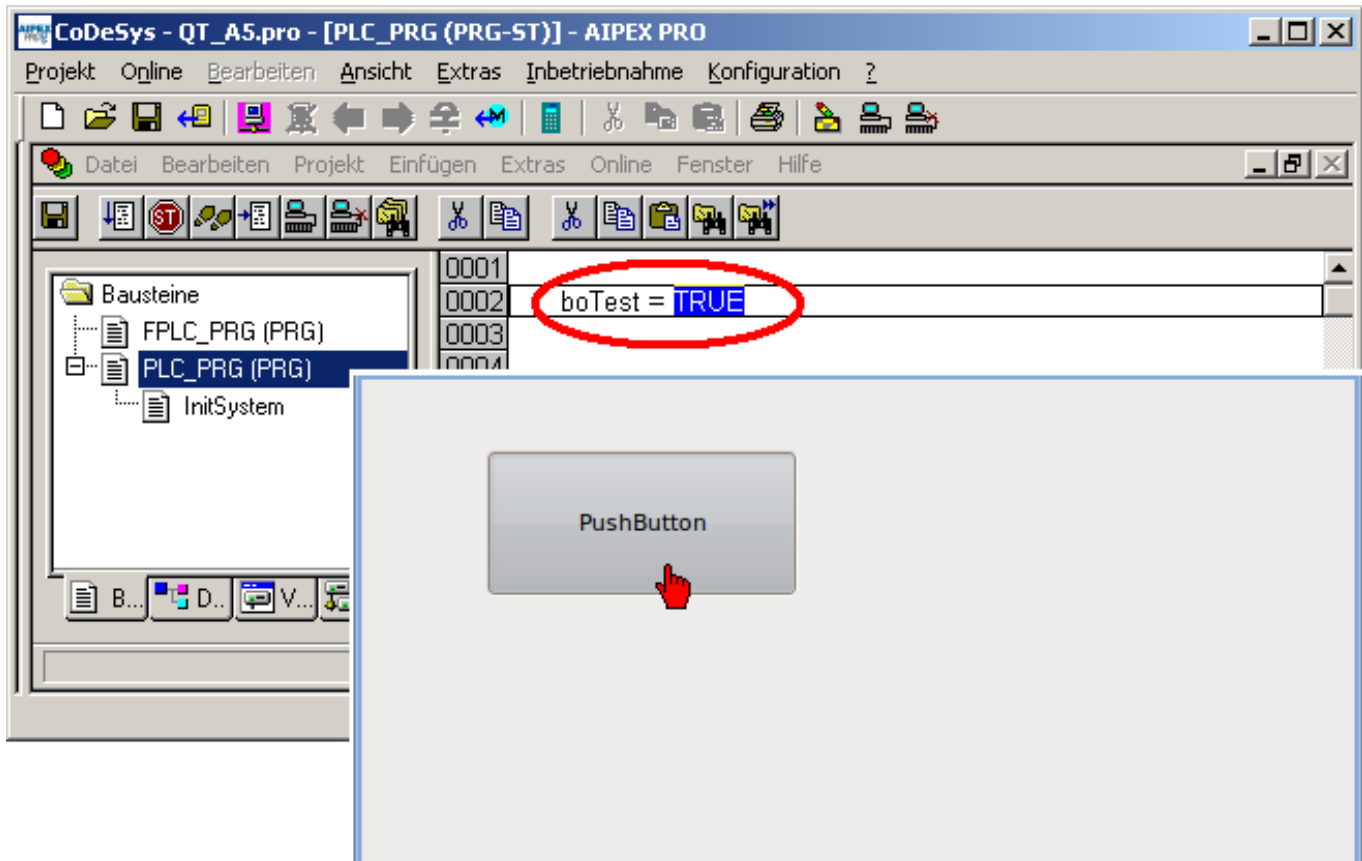
Die automatisch generierten Methoden / Funktionen / Slots dürfen nicht manuell gelöscht bzw. verändert werden, da sonst die Funktionalität nicht mehr gewährleistet ist. Es kann zu Fehlern oder undefiniertem Verhalten des Visualisierungsprogramms kommen, die Methoden werden beim nachträglichen Konfigurieren überschrieben.

5 First Steps CODESYS V2

5.1 Beschreibung First Steps

In dieser Beschreibung wird Schritt für Schritt erklärt wie Sie ein Button (toggleInd) im 'Qt Creator' mit einer CODESYS Variable (boTest) verknüpfen.

Verwendet wurde die AIPLEX PRO Version 1.08 und AMK Qt Visualisierung Version 1.03.

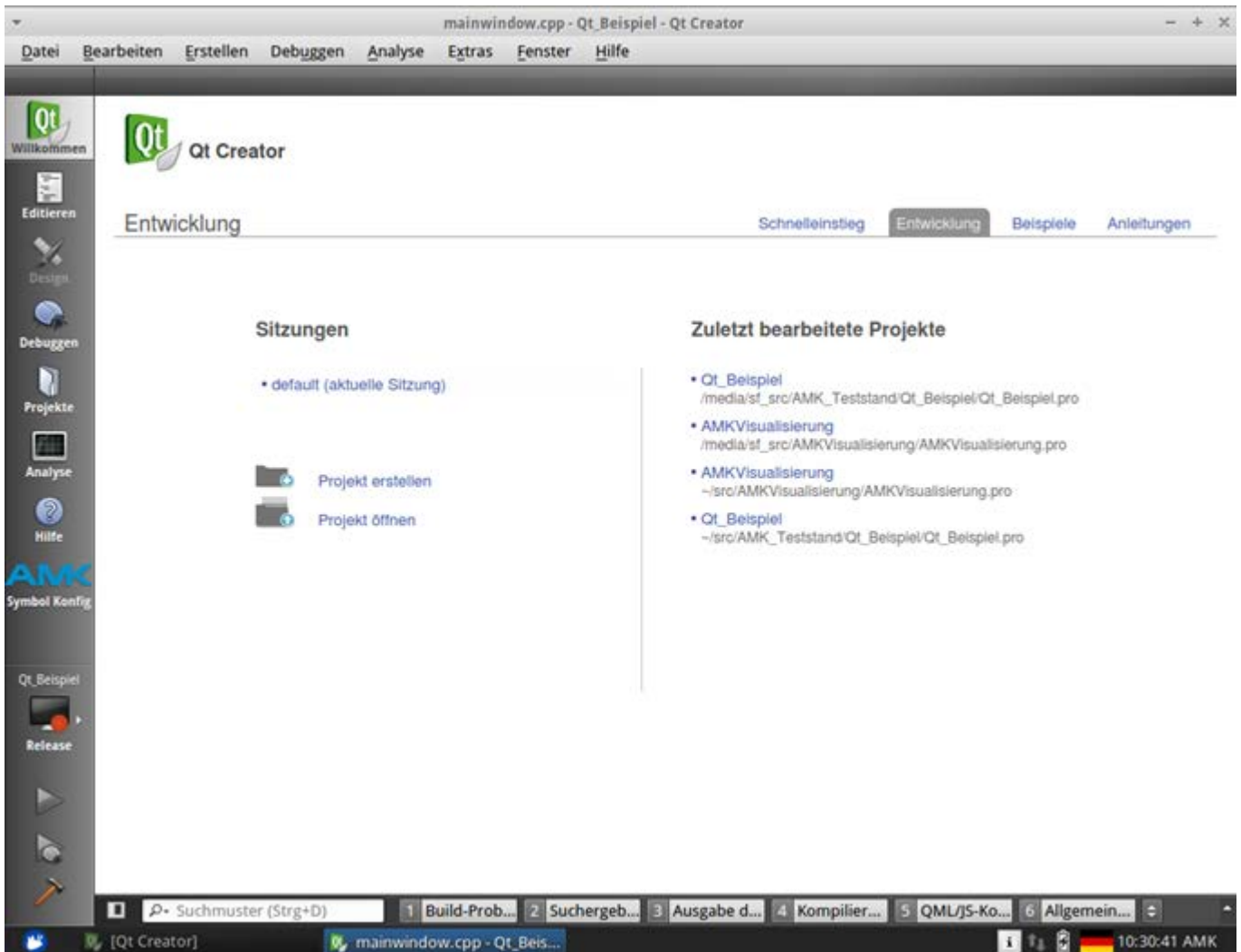


Ablauf:

- Variable anlegen und Symboldatei erzeugen
- Qt Projekt erstellen und Symboldatei einlesen
- Qt Projekt in das CODESYS Projekt integrieren

5.2 Grundeinstellungen Qt Creator

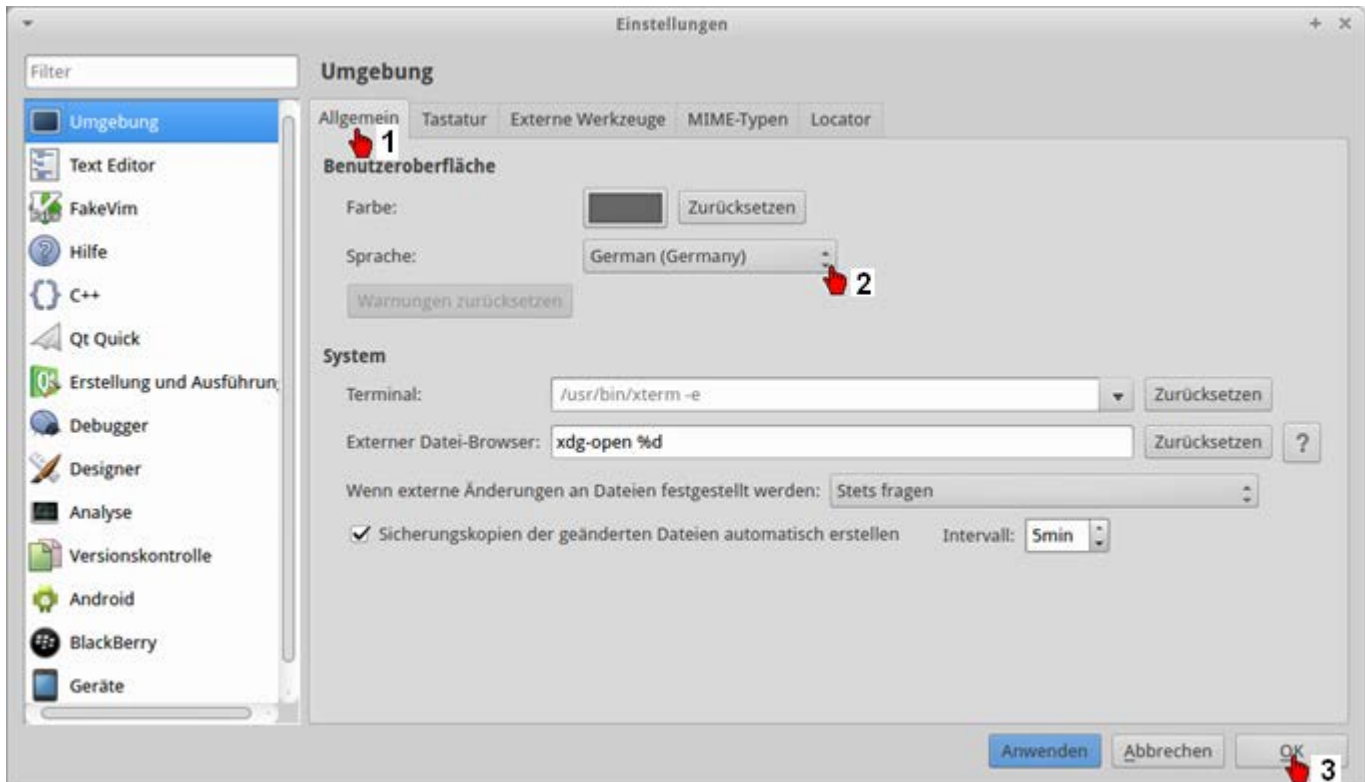
Sprache einstellen



Menü 'Tools' -> 'Options ...' anwählen

Menü 'Extras' -> 'Einstellungen' anwählen

Optionen



Anwahl Reiter **'General'**

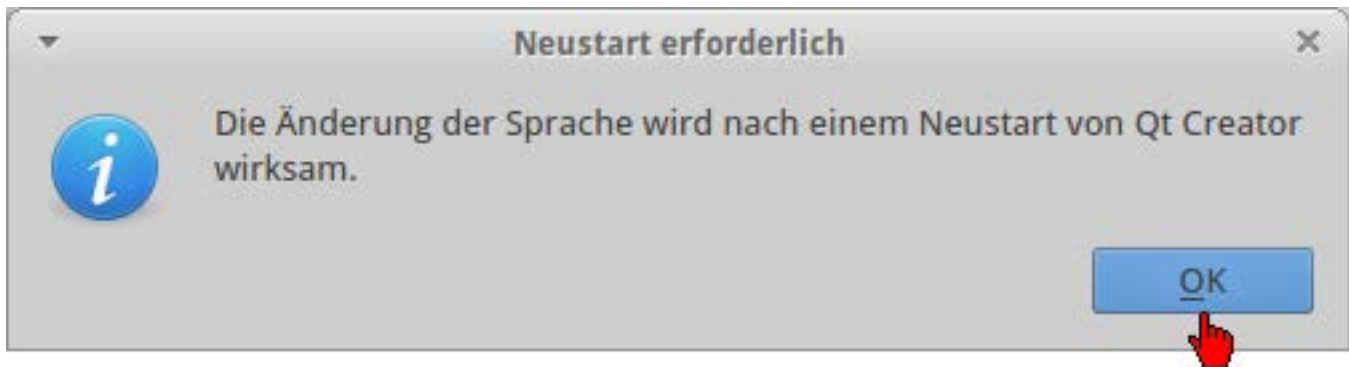
Dropdown Menü **'Language'** **'German'** anwählen

Anwahl Reiter **'Allgemein'**

Dropdown Menü **'Sprache'** **'German'** anwählen

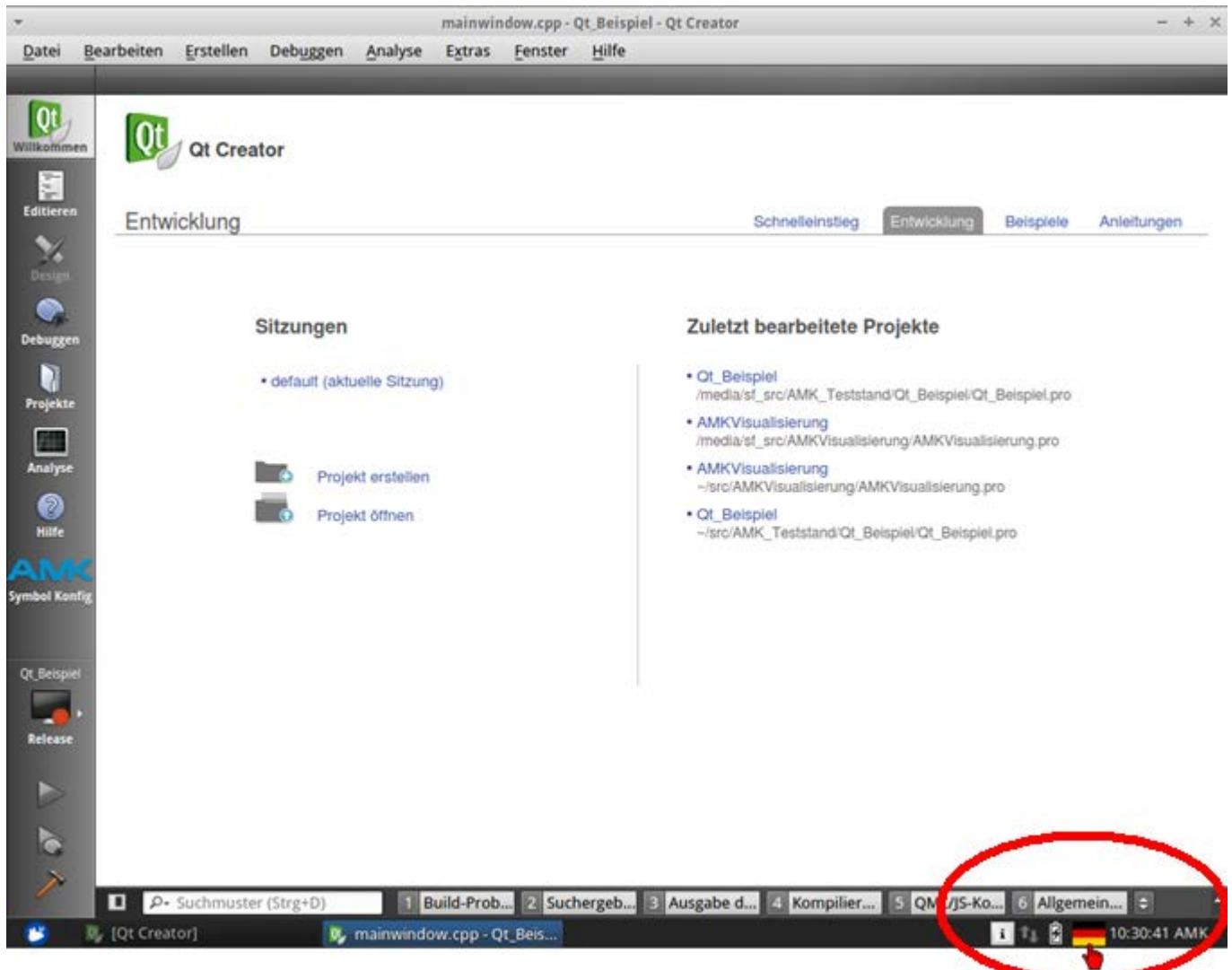
Button **'OK'** betätigen

Dialog



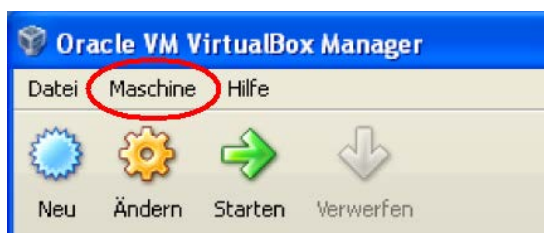
Button **'OK'** betätigen

Tastaturlayout anpassen



Flaggen-Symbol anwählen -> Flagge wechselt zwischen deutsch und englisch

Virtuelle Maschine beenden



Menü **'Maschine'** **'Schließen ...'** anwählen

Im folgenden Dialog die Option **'den Zustand der virtuellen Maschine speichern'** anwählen und mit Button **'OK'** bestätigen.
Die 'Virtuelle Maschine' wird geschlossen

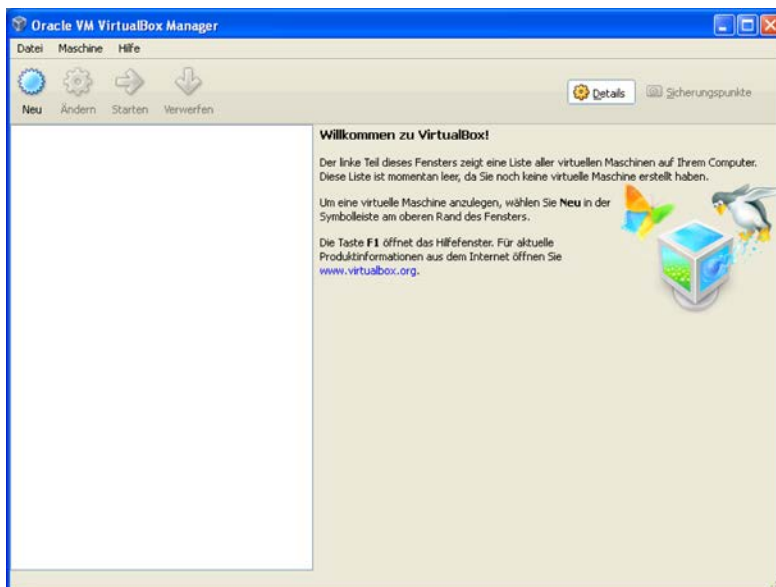


5.3 Grundeinstellungen Oracle VirtualBox

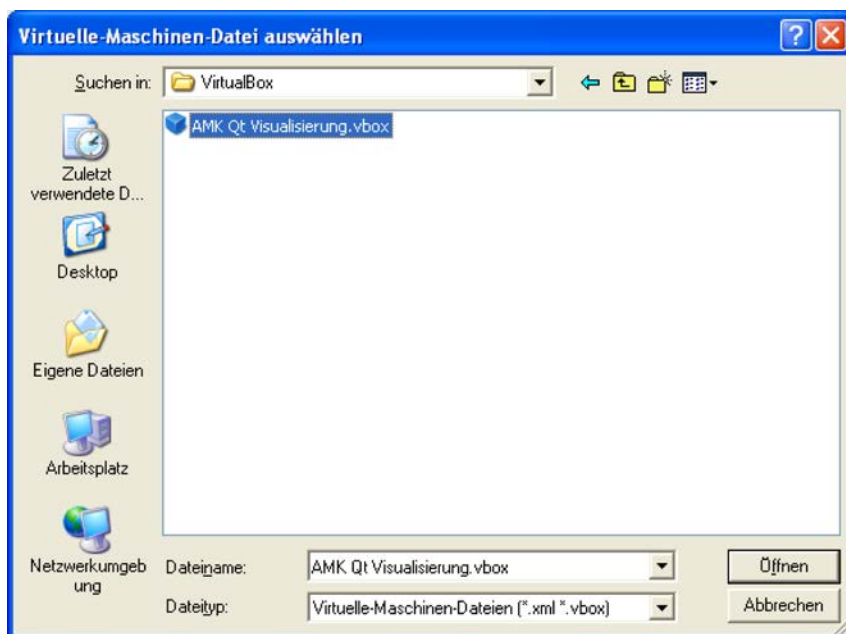
Starten



Das Symbol **Oracle VM VirtualBox** auf dem Desktop anwählen oder den entsprechenden Menüeintrag aus dem Startmenü betätigen.

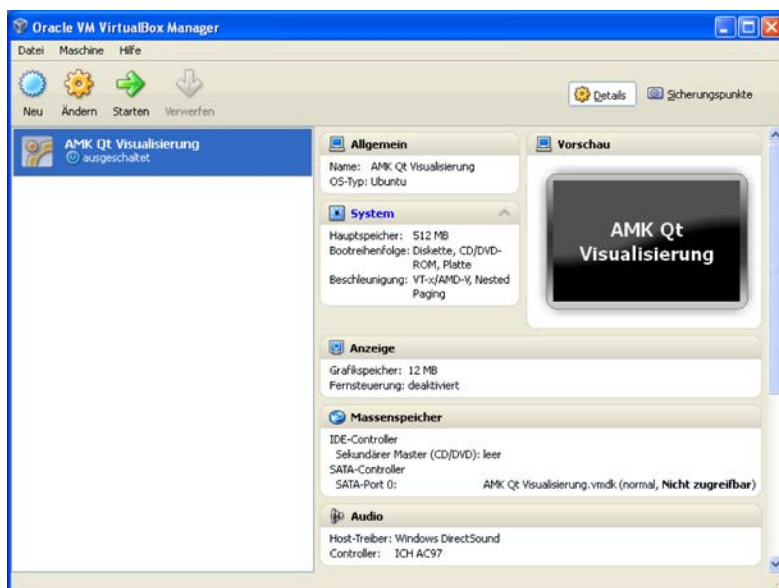


Menü **'Maschine' 'Hinzufügen ...'** aufrufen



Öffnen Sie AMK Qt Visualisierung.vbox unter
'...Eigene Dateien\VirtualBox\AMK Qt Visualisierung.vbox' mit dem Button **'Öffnen'**.

Virtuelle Maschine starten



Menü **'Maschine' 'Starten'**



Die 'Virtuelle Maschine' wird mit Linux gestartet.
Der Qt Creator wird automatisch im virtuellen Linux gestartet.
Dies kann mehrere Minuten in Anspruch nehmen.

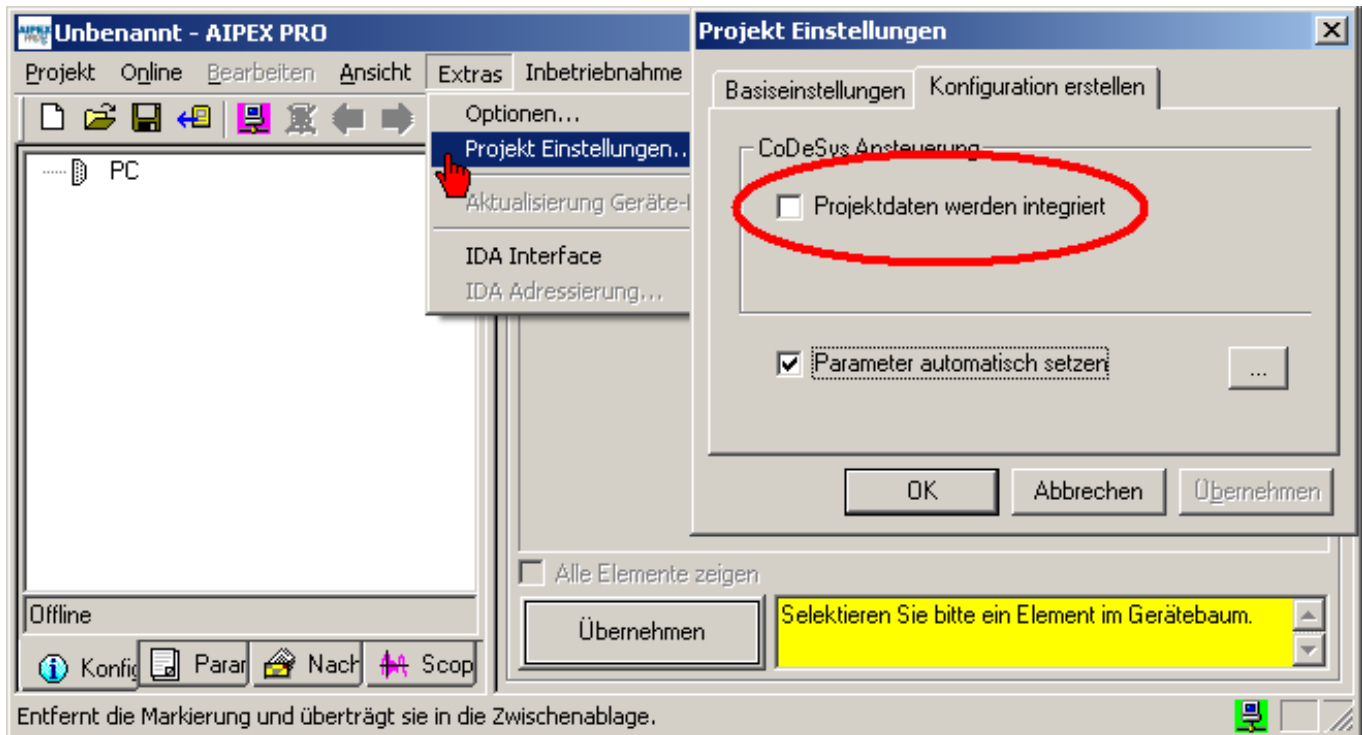
5.4 AIPEX PRO und CODESYS

5.4.1 Grundeinstellungen für AIPEX PRO

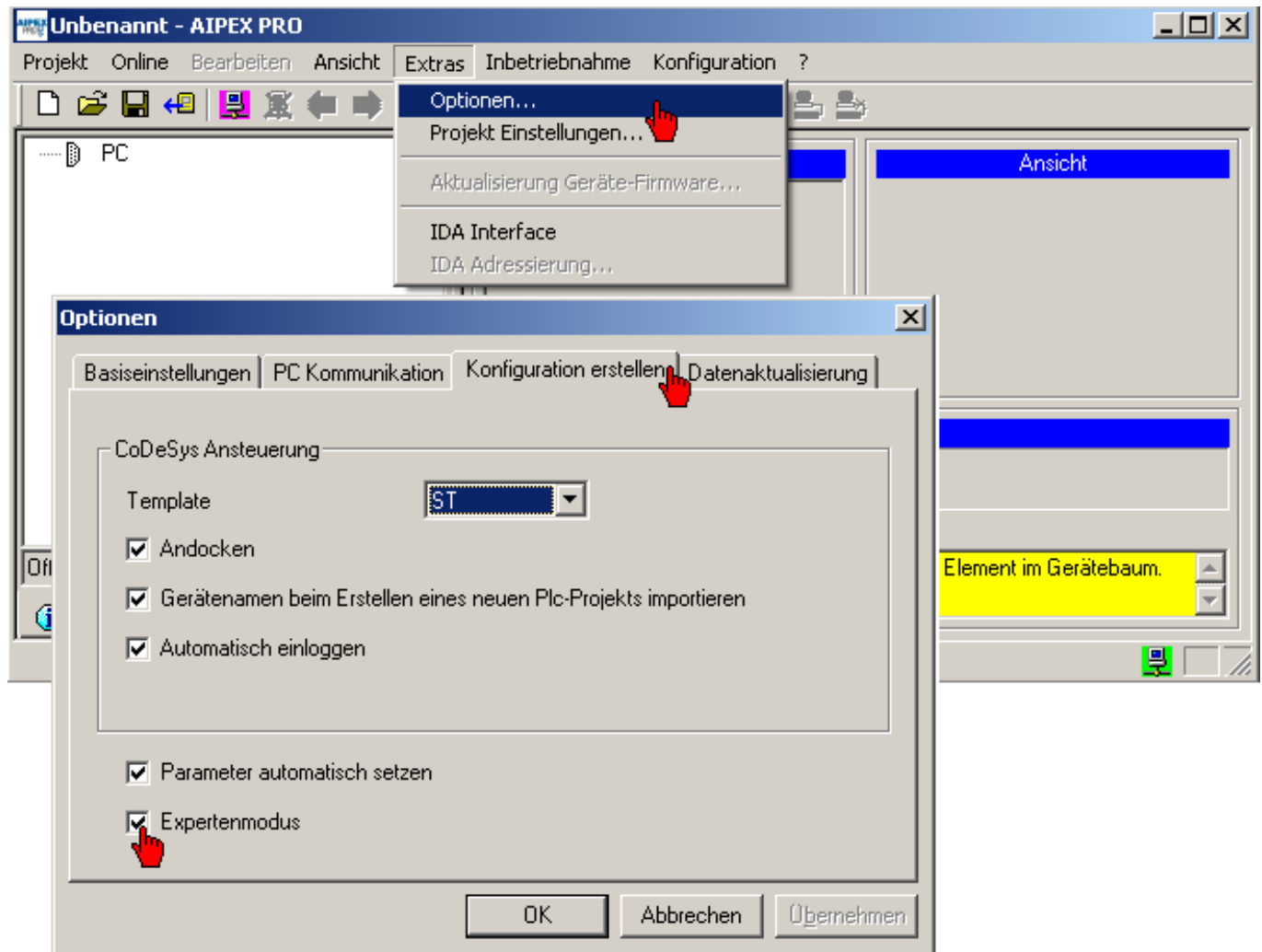
Folgende Grundeinstellungen wurden für das Qt_Beiispiel Projekt verwendet.

Start Sie **'AIPEX PRO'**.

Wählen Sie **'Projektdaten werden integriert'** unter **'Extras' 'Projekteinstellungen'** ab.



Blenden Sie das CODESYS Menü 'Online' 'Einloggen' ein, indem Sie unter '**Extras**' '**Optionen**' '**Konfiguration erstellen**' den '**Expertenmodus**' aktivieren.



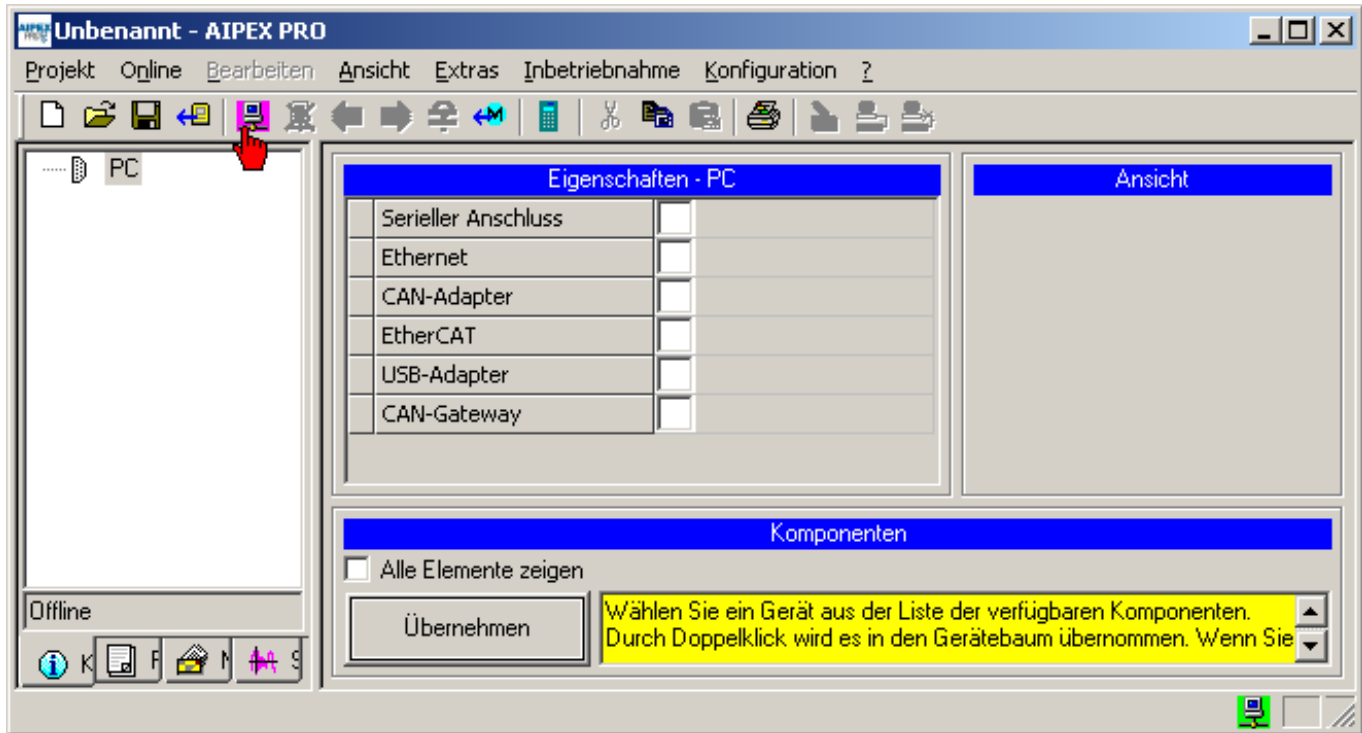
Führen Sie anschließend einen Neustart von 'AIPEX PRO' durch.

5.4.2 APEX PRO Projekt erzeugen

In diesem Kapitel wird beschrieben, wie Sie ein APEX PRO Projekt erzeugen und speichern.

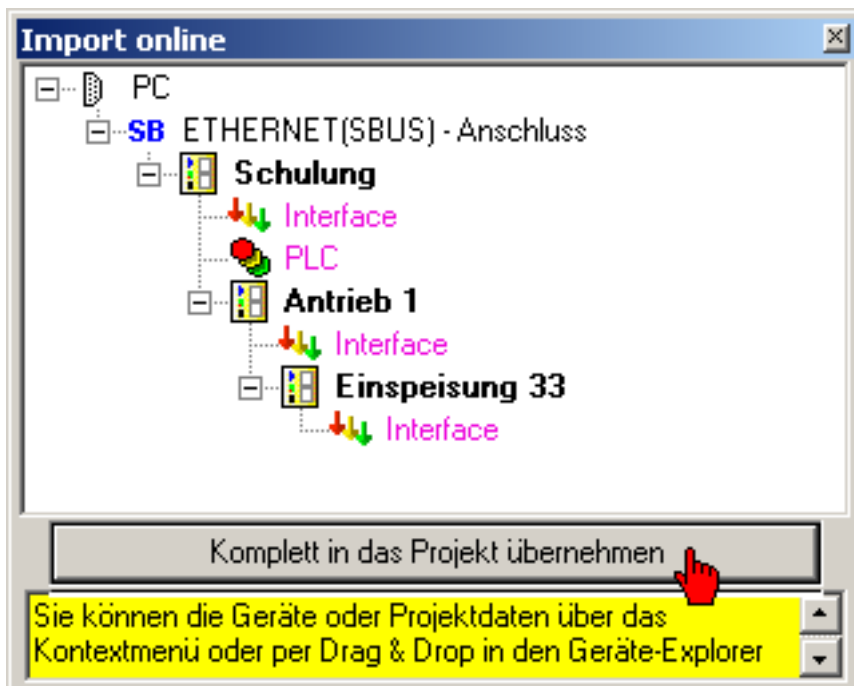
Start Sie **'APEX PRO'**.

Klicken Sie auf den Taster **'Einloggen'**.



Übernehmen Sie Ihr Projekt in den 'APEX PRO Gerätebaum'.

Klicken Sie dazu den Taster **'Komplett in das Projekt übernehmen'**.

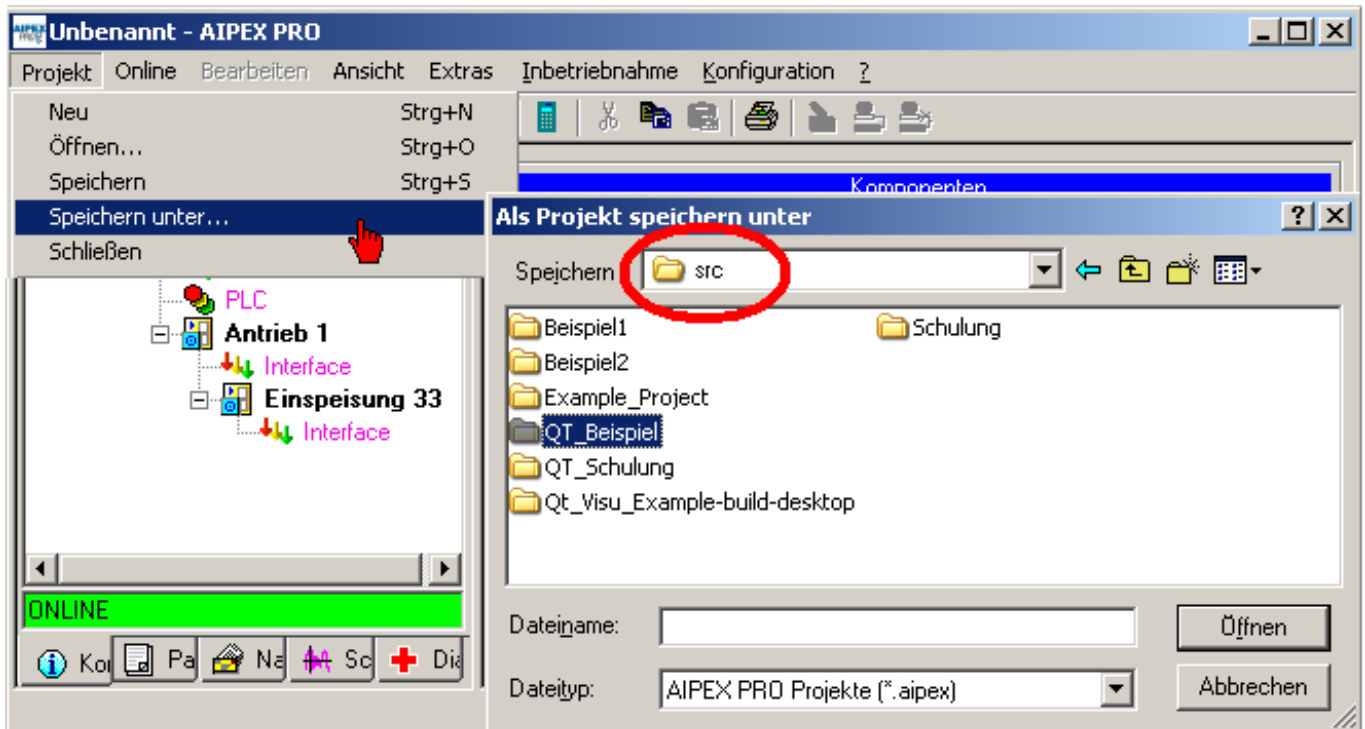




Das AIPEX PRO Projekt muss im automatisch erzeugten Ordner 'C:\AMK\src\...' gespeichert werden. Auf diesen Ordner kann die 'Virtuelle Maschine' standardmäßig zugreifen.

Anwenderspezifische Ordner [Siehe Anwenderspezifischer Ordner auf Seite 75.](#)

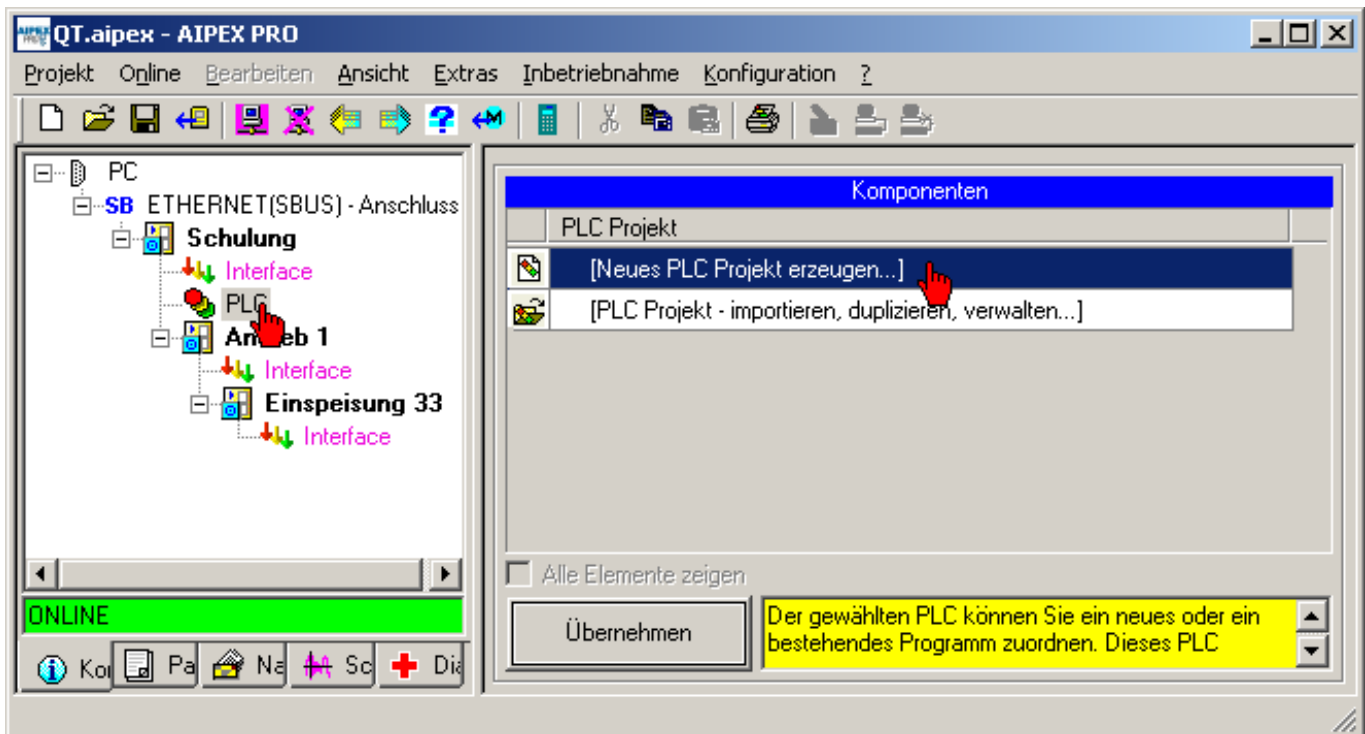
In diesem Beispiel wird der nachfolgend der Name Qt_Beispiel verwendet.



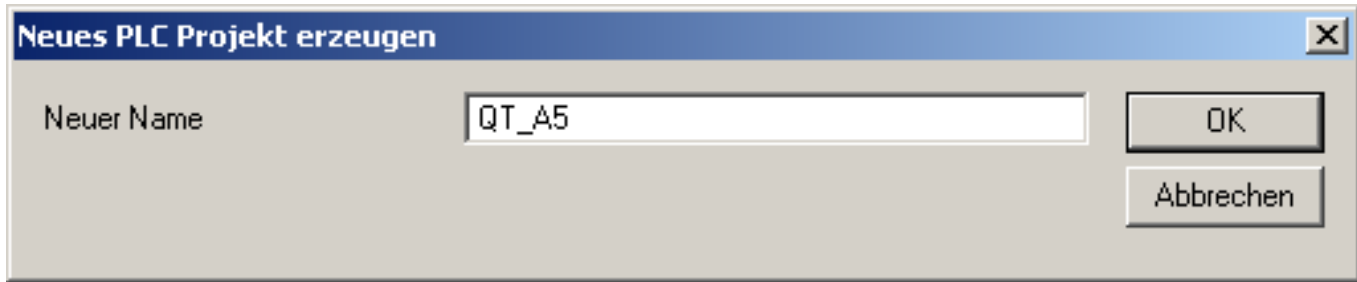
5.4.3 PLC Projekt erzeugen

In diesem Kapitel wird beschrieben, wie Sie ein PLC Projekt erzeugen. Das PLC Projekt wird automatisch gespeichert.

Klicken Sie im Gerätebaum auf 'PLC' und anschließend unter dem Menü 'Komponenten' auf 'Neues PLC Projekt erzeugen'.



Vergeben Sie einen Projektnamen für Ihr PLC Projekt.
In diesem Beispiel wird der nachfolgend der Name QT_A5 verwendet.

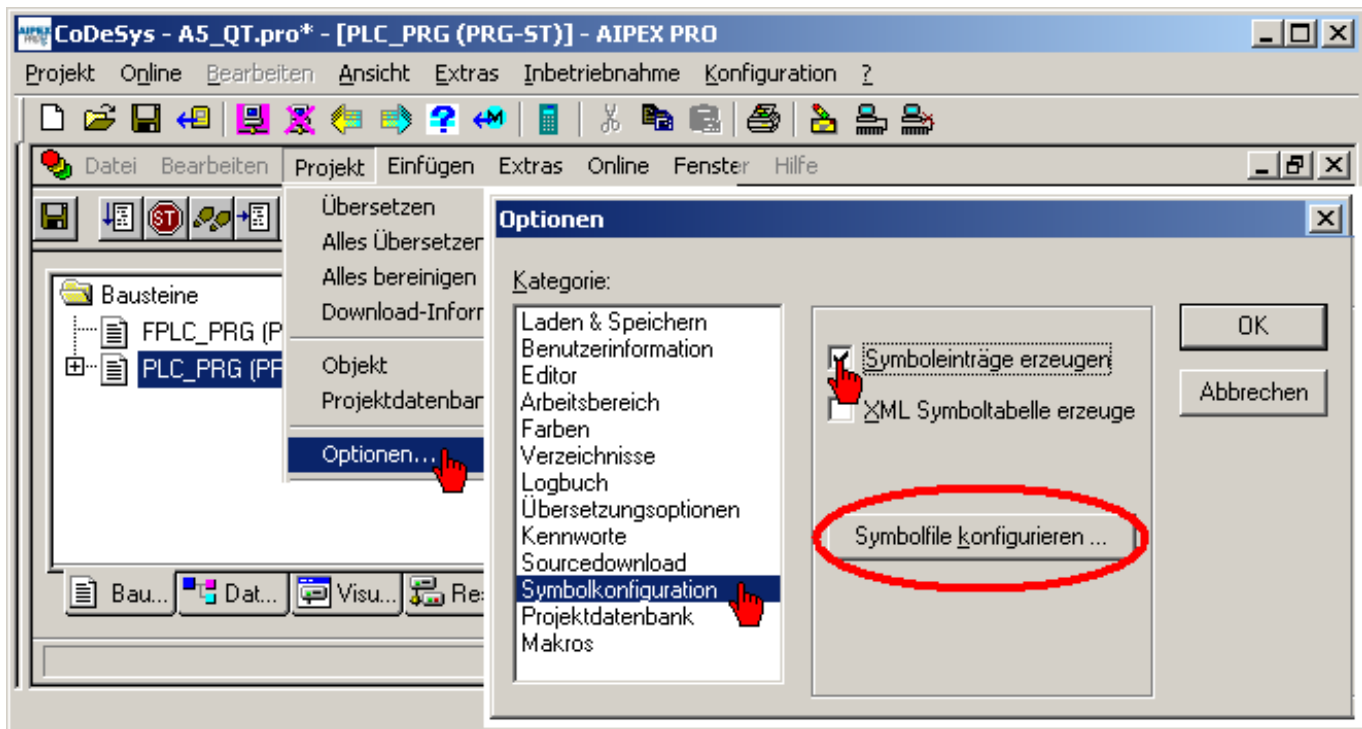


5.4.4 Grundeinstellungen CODESYS (Symboldatei)

Die Symboldatei wird für den Datenaustausch zwischen 'AIPEX PRO' und der 'Qt Visualisierung' benötigt. Durch das Aktivieren der Funktion 'Symboldatei senden' wird die beim Übersetzen, automatisch erzeugte Symboldatei beim Download in die Steuerung geladen.

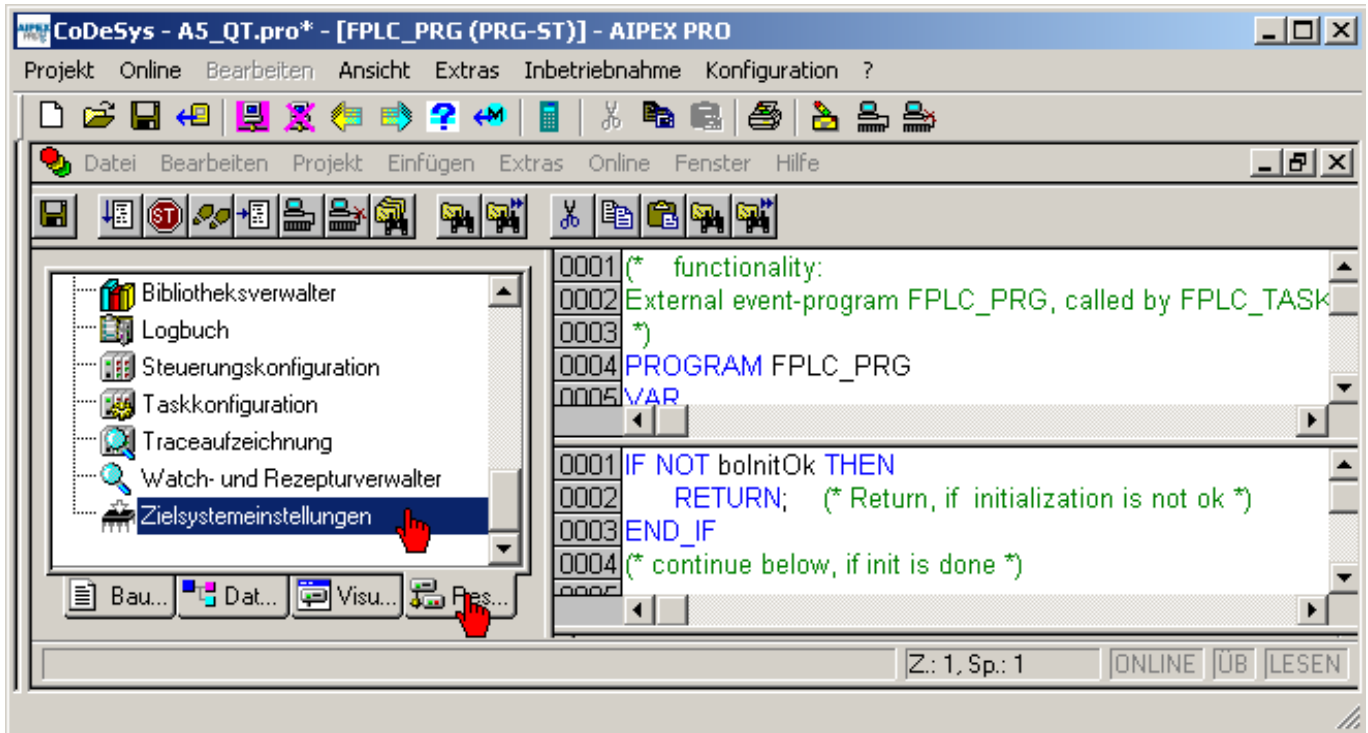
Symboldatei erzeugen:

Aktivieren Sie **'Symboleinträge erzeugen'** unter dem Menüpunkt **'Projekt' 'Optionen' 'Symbolkonfiguration'**.
Die Einträge (PLC Variablen) der Symboldatei werden durch die Funktion **'Symbolfile konfigurieren'** ausgewählt.

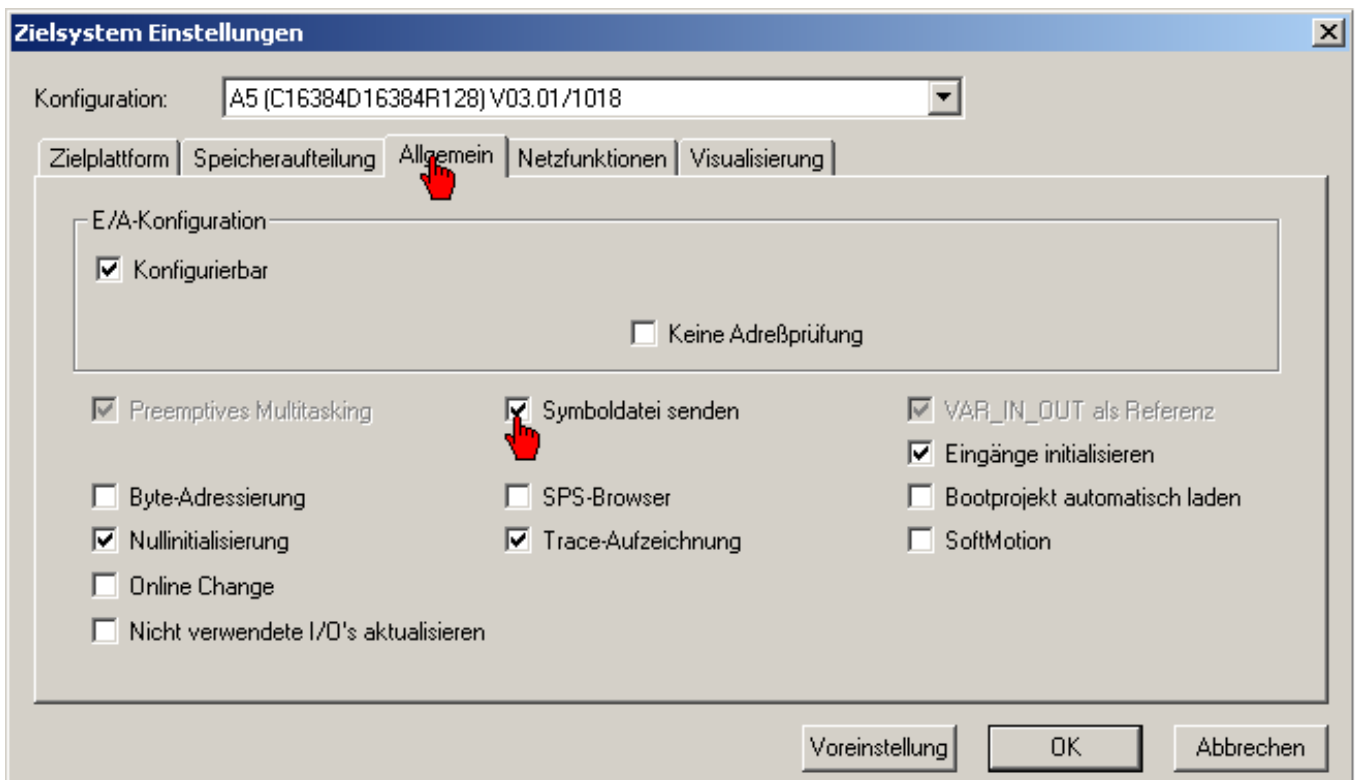


Symboldatei senden:

Öffnen Sie die **'Zielsystem Einstellungen'** unter dem Reiter **'Ressourcen'**.



Aktivieren Sie unter dem Reiter **'Allgemein'**, **'Symboldatei senden'**.

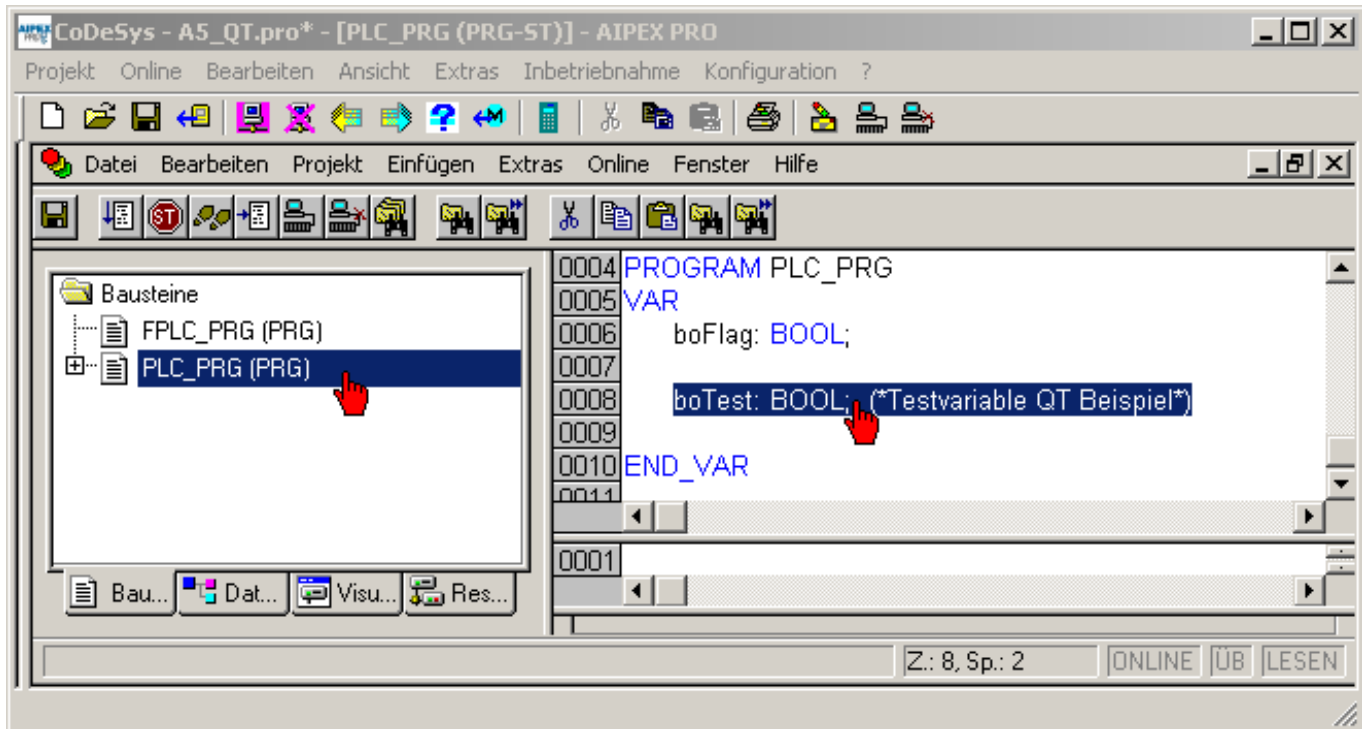


5.4.5 Testvariable anlegen

In diesem Kapitel wird beschrieben, wie Sie die Testvariable <boTest> in CODESYS anlegen. Diese Variable soll anschließend über die 'Qt Visualisierung' gesetzt werden.

Wechseln Sie in den asynchronen Programmbaustein '**PLC_PRG**'.

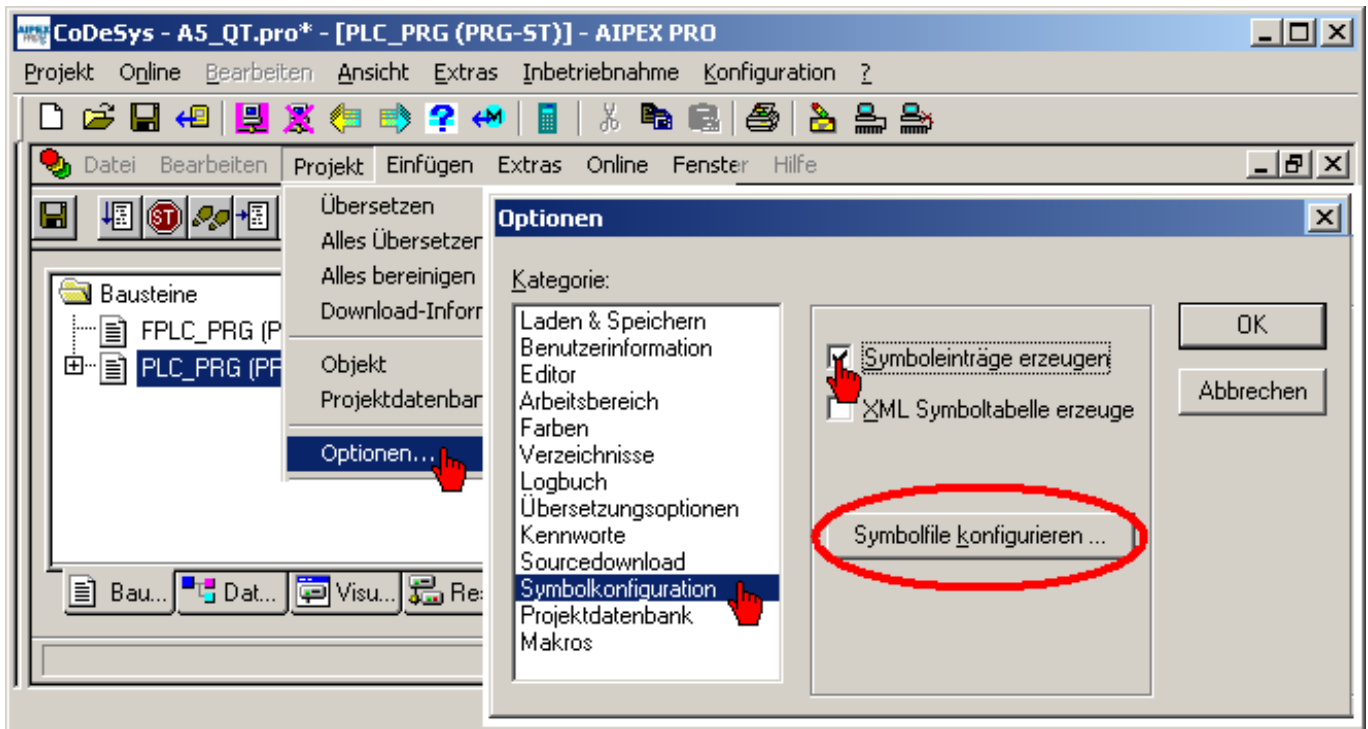
Legen Sie im Deklarationsfenster eine Variable <boTest:BOOL;> an.



5.4.6 Symboldatei konfigurieren

In diesem Kapitel wird beschrieben, wie Sie die Symboldatei für den Datenaustausch zwischen PLC Programm und 'Qt Creator' konfigurieren. Im Beispiel wird die Variable <boTest> eingefügt.

Klicken Sie auf den Taster **'Symbolfile konfigurieren'** unter dem Menüpunkt **'Projekt' 'Optionen' 'Symbolkonfiguration'**.



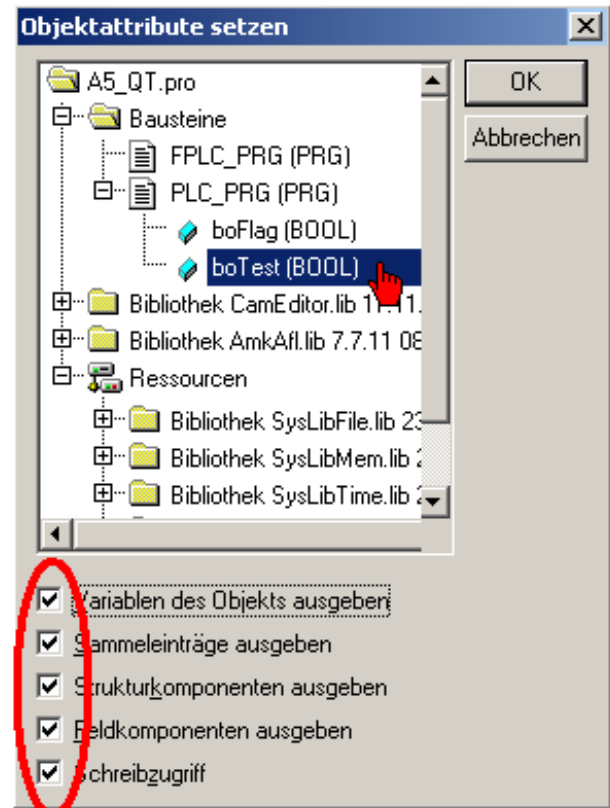
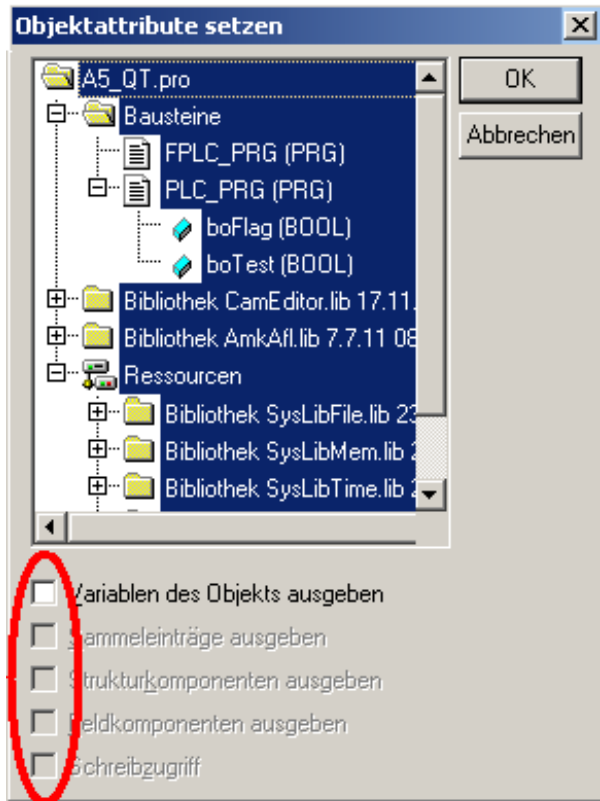


Deaktivieren Sie erst alle Kontrollhäkchen. Wählen Sie anschließend Ihre Variable(en) aus die Sie zwischen 'AIPEX PRO' und 'Qt Visualisierung' austauschen möchten. Danach müssen Sie alle Kontrollhäkchen wieder setzen.

Beim nächsten Aufruf der Funktion 'Symbolfile konfigurieren' müssen Sie nur die neu dazugekommen Variablen anwählen. Danach müssen Sie alle Kontrollhäkchen wieder setzen.

Werden mehrere Variablen gleichzeitig markiert, so werden diese Einstellungen für alle markierten Variablen übernommen.

Nachdem die Einstellungen vorgenommen wurden, schließen Sie die Fenster "Objektattribute setzen" und "Optionen" jeweils durch einen Klick auf die Schaltfläche "OK".

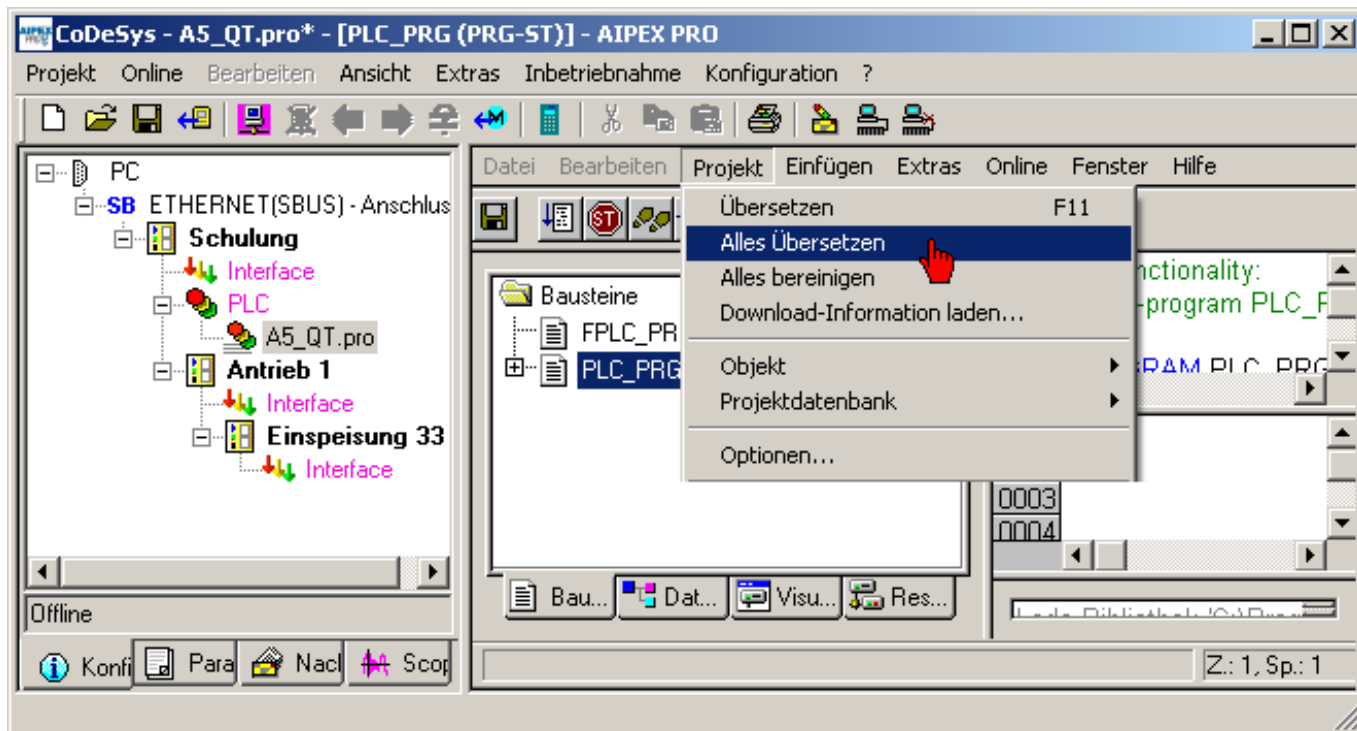


5.4.7 Symboldatei erzeugen

In diesem Kapitel wird beschrieben, wie Sie die Symboldatei für den Datenaustausch erzeugen. Die Symboldatei PLC Programmname.SYM (im Beispiel QT_A5.SYM) wird im AIPEX PRO Projekt im Ordner 'PLC' gespeichert.

Die Symboldatei wird automatisch beim Übersetzen des PLC Programm erzeugt.

Z.B. durch **'Projekt' 'Alles Übersetzen'**.



5.5 Qt Creator

5.5.1 Wizard 'Neues Qt Projekt' erzeugen

In diesem Kapitel wird beschrieben, wie Sie mit dem 'AMK Projekt Wizard' ein neues Qt Projekt anlegen.

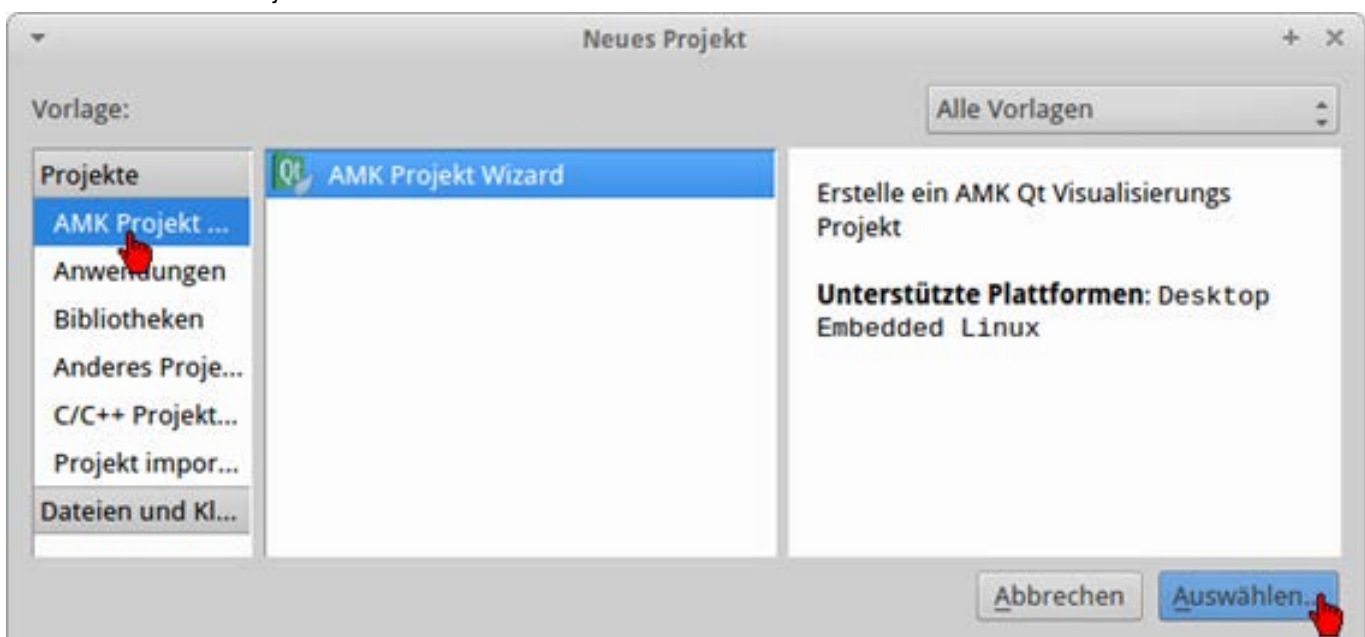
Starten Sie den 'Qt Creator'.



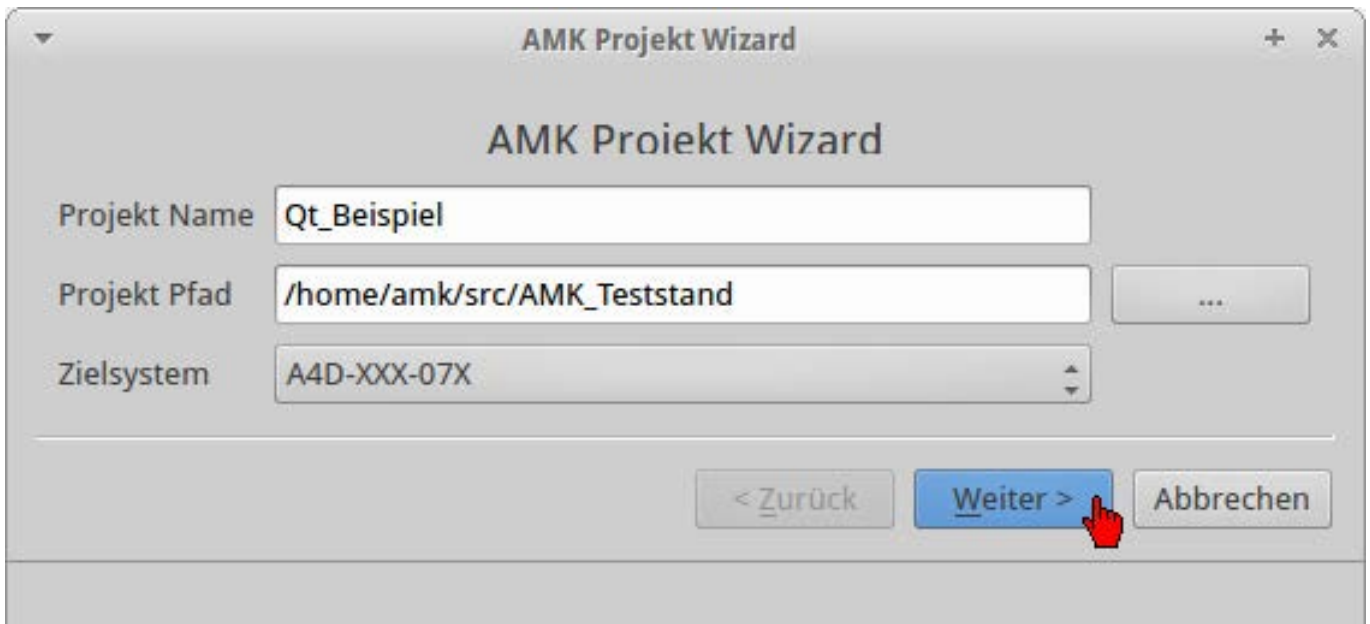
Starten Sie ein neues Projekt wird mit der Taste **'Projekt erstellen'**.



Starten Sie den 'AMK Projekt Wizard' mit der Taste **'Auswählen'**.

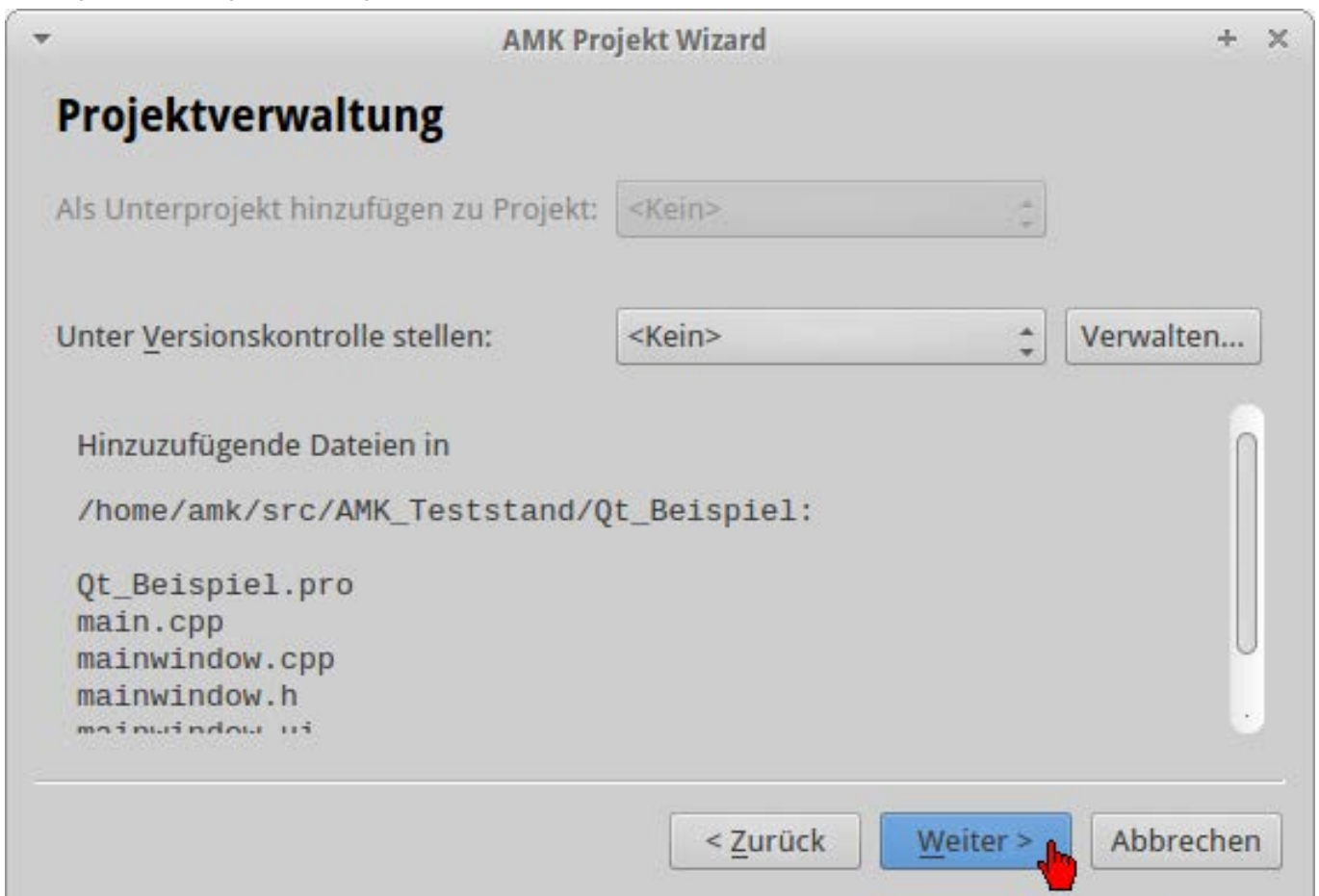


Geben Sie einen 'Projekt Namen',
den 'Projekt Pfad' (Ordner wo Sie Ihr AIPEX PRO Projekt gespeichert haben)
und Ihr 'Zielsystem' (verwendete AMK Steuerung) ein.
Bestätigen Sie mit der Taste **'Weiter'**.



The image shows the 'AMK Projekt Wizard' dialog box. It has a title bar with a dropdown arrow, a plus icon, and a close icon. The main title is 'AMK Projekt Wizard'. There are three input fields: 'Projekt Name' with the text 'Qt_Beispiel', 'Projekt Pfad' with the text '/home/amk/src/AMK_Teststand' and a browse button '...', and 'Zielsystem' with a dropdown menu showing 'A4D-XXX-07X'. At the bottom, there are three buttons: '< Zurück', 'Weiter >' (highlighted with a red hand cursor), and 'Abbrechen'.

Bestätigen Sie den folgenden Dialog mit dem Taster **'Abschließen'**.

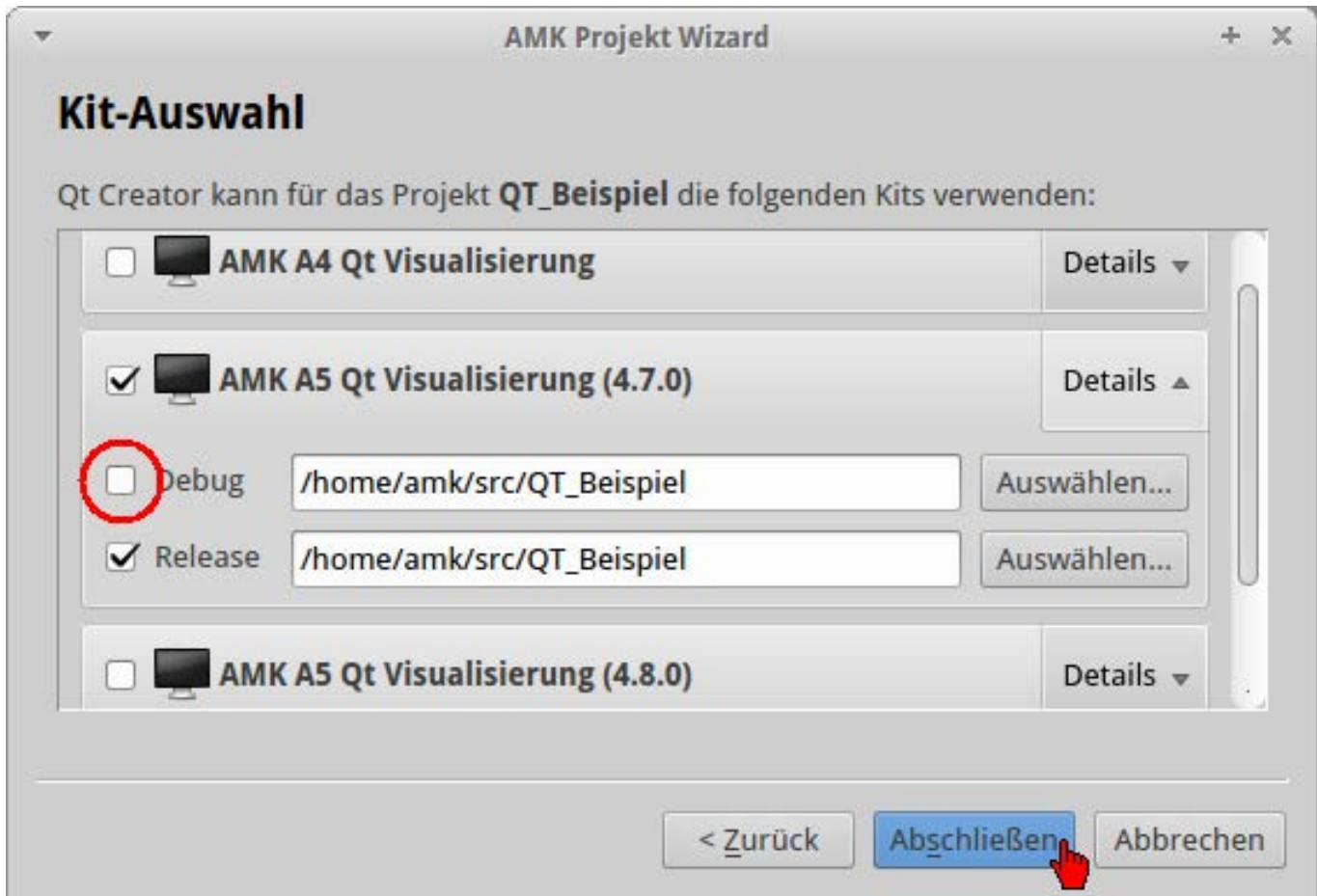


The image shows the 'AMK Projekt Wizard' dialog box in the 'Projektverwaltung' (Project Management) step. The title bar is the same. The main title is 'Projektverwaltung'. There are two dropdown menus: 'Als Unterprojekt hinzufügen zu Projekt:' with the text '<Kein>' and 'Unter Versionskontrolle stellen:' with the text '<Kein>'. There is a 'Verwalten...' button next to the second dropdown. Below these, there is a section titled 'Hinzuzufügende Dateien in' followed by the path '/home/amk/src/AMK_Teststand/Qt_Beispiel:'. Below the path, there is a list of files: 'Qt_Beispiel.pro', 'main.cpp', 'mainwindow.cpp', 'mainwindow.h', and 'mainwindow.ui'. At the bottom, there are three buttons: '< Zurück', 'Weiter >' (highlighted with a red hand cursor), and 'Abbrechen'.

Wählen Sie das Menü **'Details'** an.

Entfernen Sie den Haken bei 'Debug'.

Durch Klicken des Tasters **'Abschließen'** wird das Menü 'Projekt einrichten' abgeschlossen.

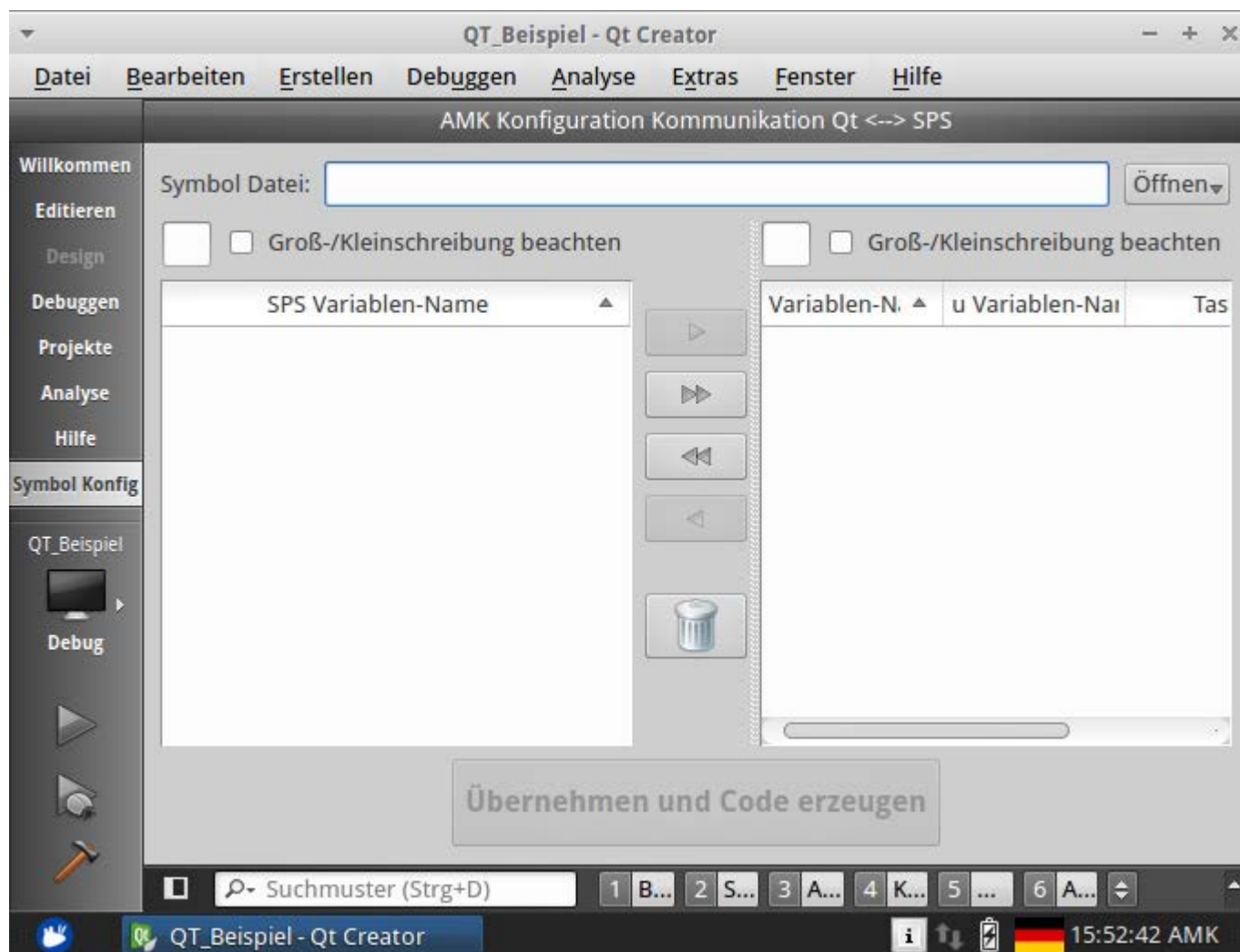


5.5.2 Datenaustausch zwischen PLC und Qt

Dieses Kapitel beschreibt, wie Sie die Symboldatei öffnen und Sie Ihre CODESYS PLC Variablen an den 'Qt Creator' übergeben.

Öffnen Sie den 'Kommunikations Qt <--> SPS Editor' durch klicken auf folgendes Symbol.

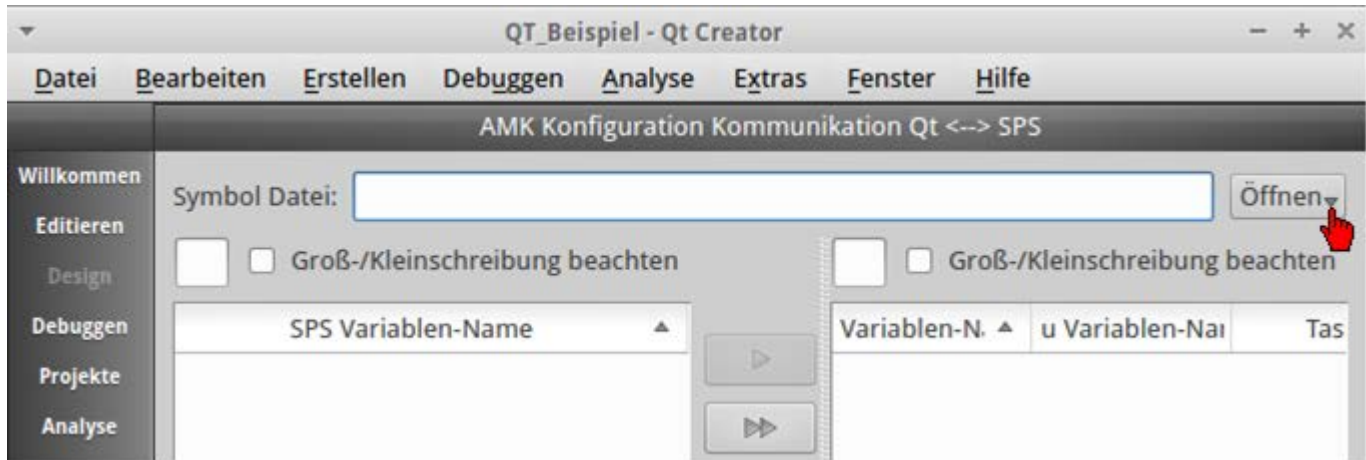
Symbol Konfig



Symboldatei laden:

Beim ersten Laden der Symboldatei müssen Sie kurz auf den Taster **'Öffnen'** klicken.

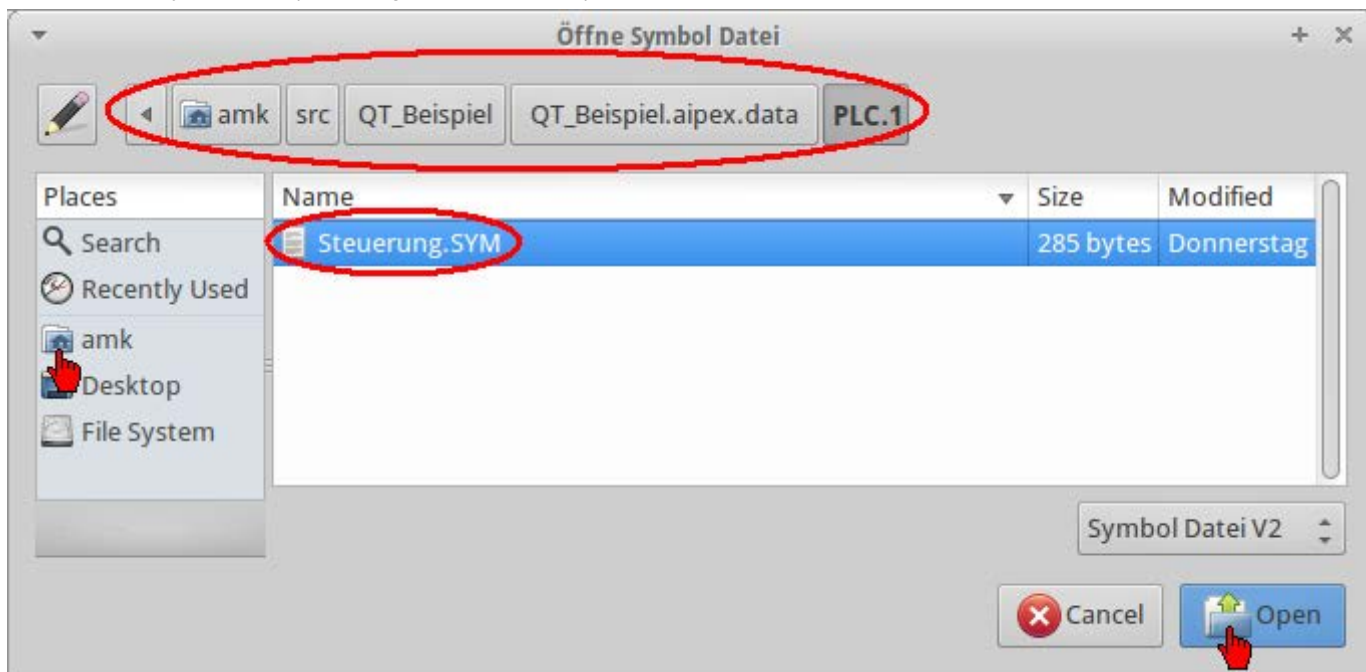
Eine bestehende Konfiguration wird verworfen. Verwenden Sie die Funktion 'Neu laden', wenn Sie Ihre Symboldatei ein weiteres Mal aufgrund einer Aktualisierung in CODESYS einlesen möchten.



Wählen Sie das Verzeichnis an, indem Sie das AIPEX PRO Projekt gespeichert haben.

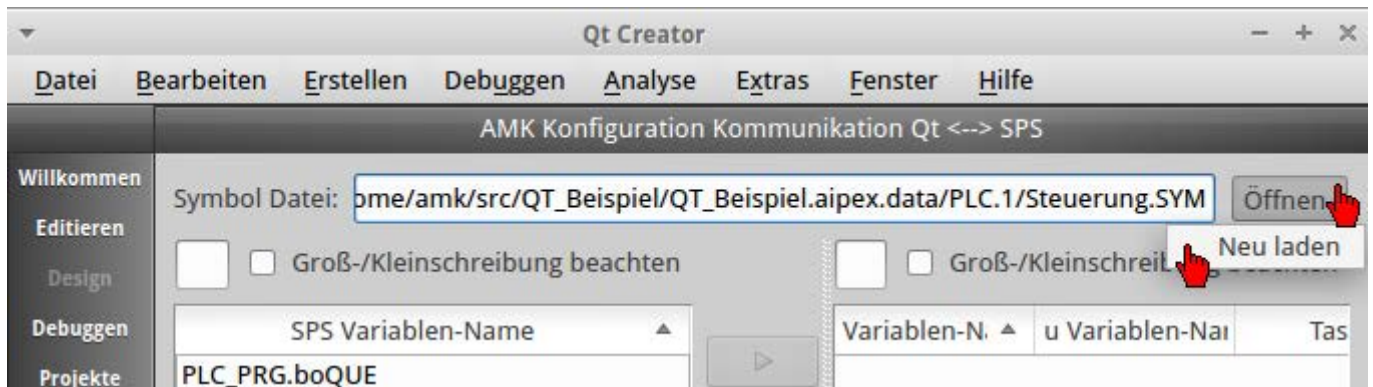
Defaultverzeichnis 'C:\AMK\src\Qt Projektname'

Sie finden die Symboldatei (PLC Programmname.SYM) im 'PLC' Ordner.



Symboldatei 'Neu laden':

Halten Sie den Taster **'Öffnen'** so lange gedrückt, bis das Menü **'Neu laden'** eingeblendet wird. Klicken Sie anschließend auf **'Neu laden'**.



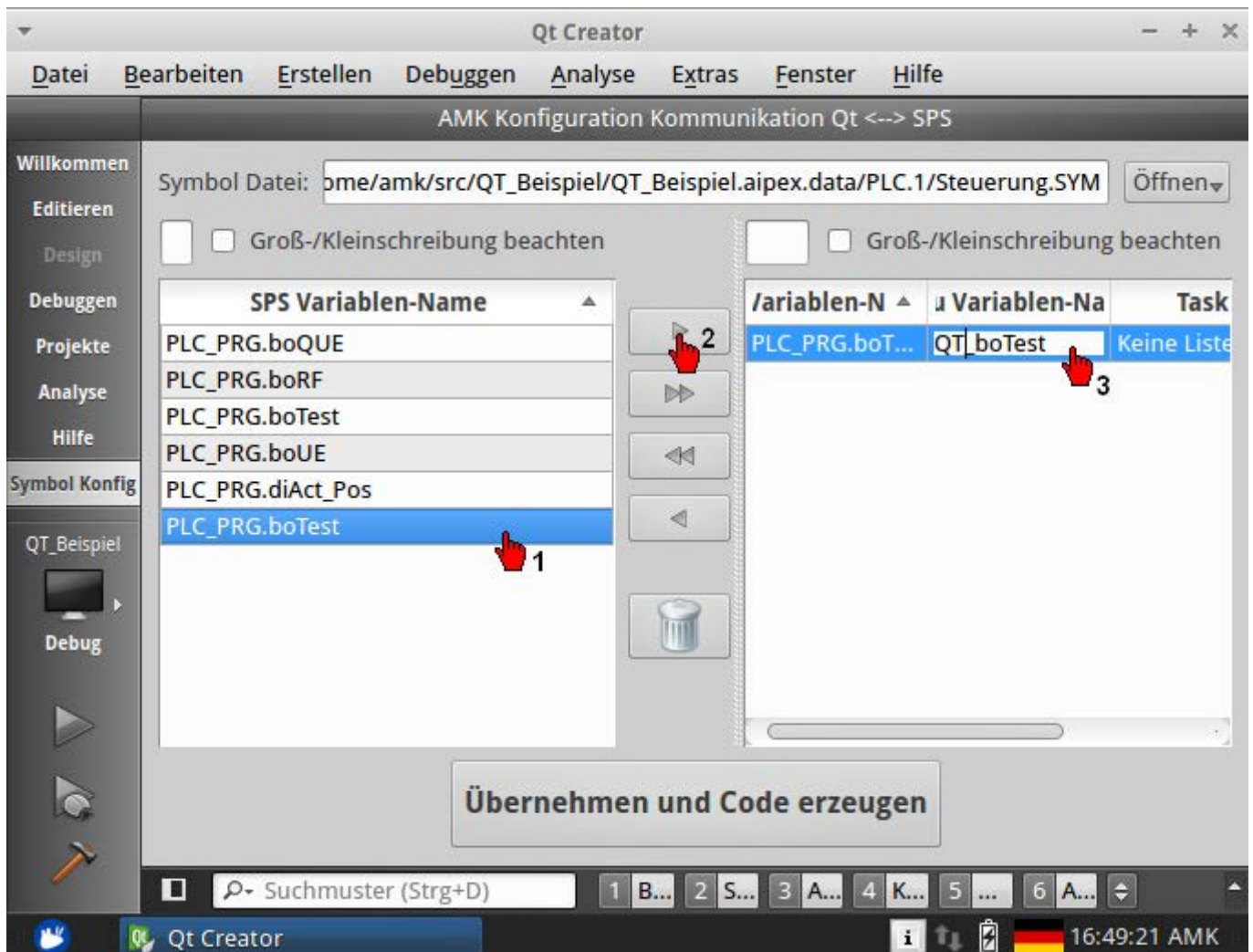
In der Linken Zeile (Inhalt Symboldatei) werden die CODESYS PLC Variablen angezeigt. Durch Doppelklick, Drag & Drop oder die Pfeil Taste werden die angewählten Variablen übernommen.

SPS Variablen-Name: original CODESYS PLC Variablen Name

Visu Variablen-Name: Dieser Name wird innerhalb des 'Qt Creators' verwendet. Sie können den Variablennamen spezifisch anpassen.

Task: Auswahl der Übertragungsart, keine Task (nur funktionaler Aufruf) oder zyklisch.

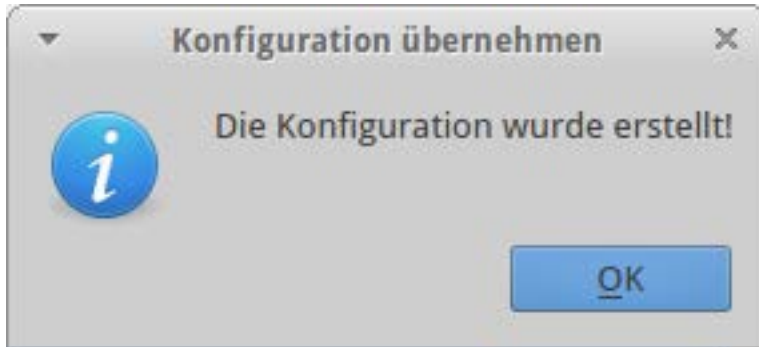
Übernehmen Sie die Testvariable 'PLC_PRG.boTest'. Anschließend können Sie die Variable umbenennen (im Beispiel 'QT_boTest').



Erzeugen Sie den 'Qt Programmcode' für die ausgewählten Variablen durch klicken auf die Taste '**Übernehmen und Code erzeugen**'.

Übernehmen und Code erzeugen

Meldung einer Fehlerfreien Code Generierung.



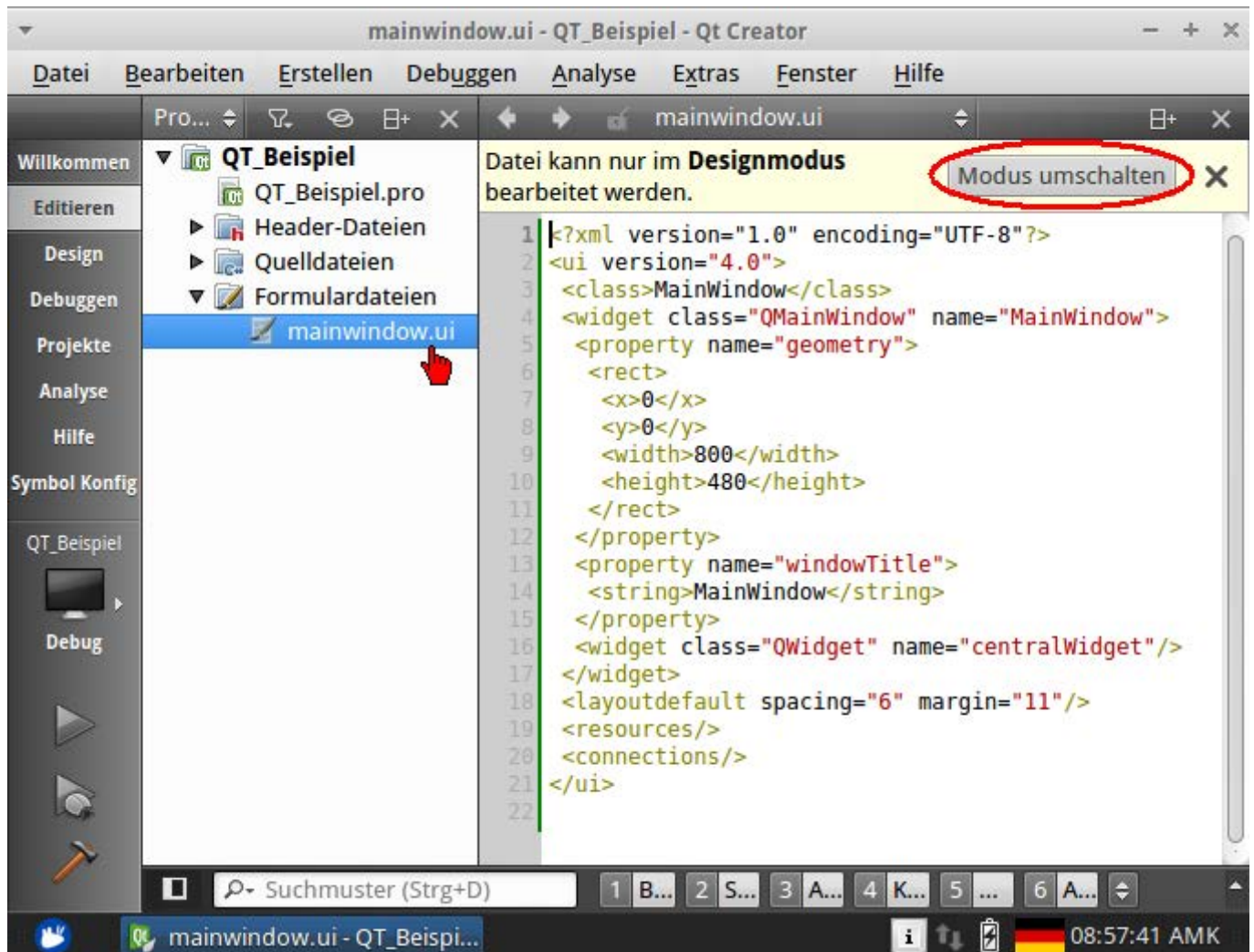
5.5.3 Qt Editor

Dieses Kapitel beschreibt, wie Sie in den Grafikeditor einen Button einfügen, umbenennen und den dazugehörigen Programmcode eingeben. Verwendet wird die in CODESYS angelegte Testvariable <boTest>, Qt Visu Name QT_boTest. Starten Sie den Grafikeditor durch klicken auf die Taste **'Editieren'**.

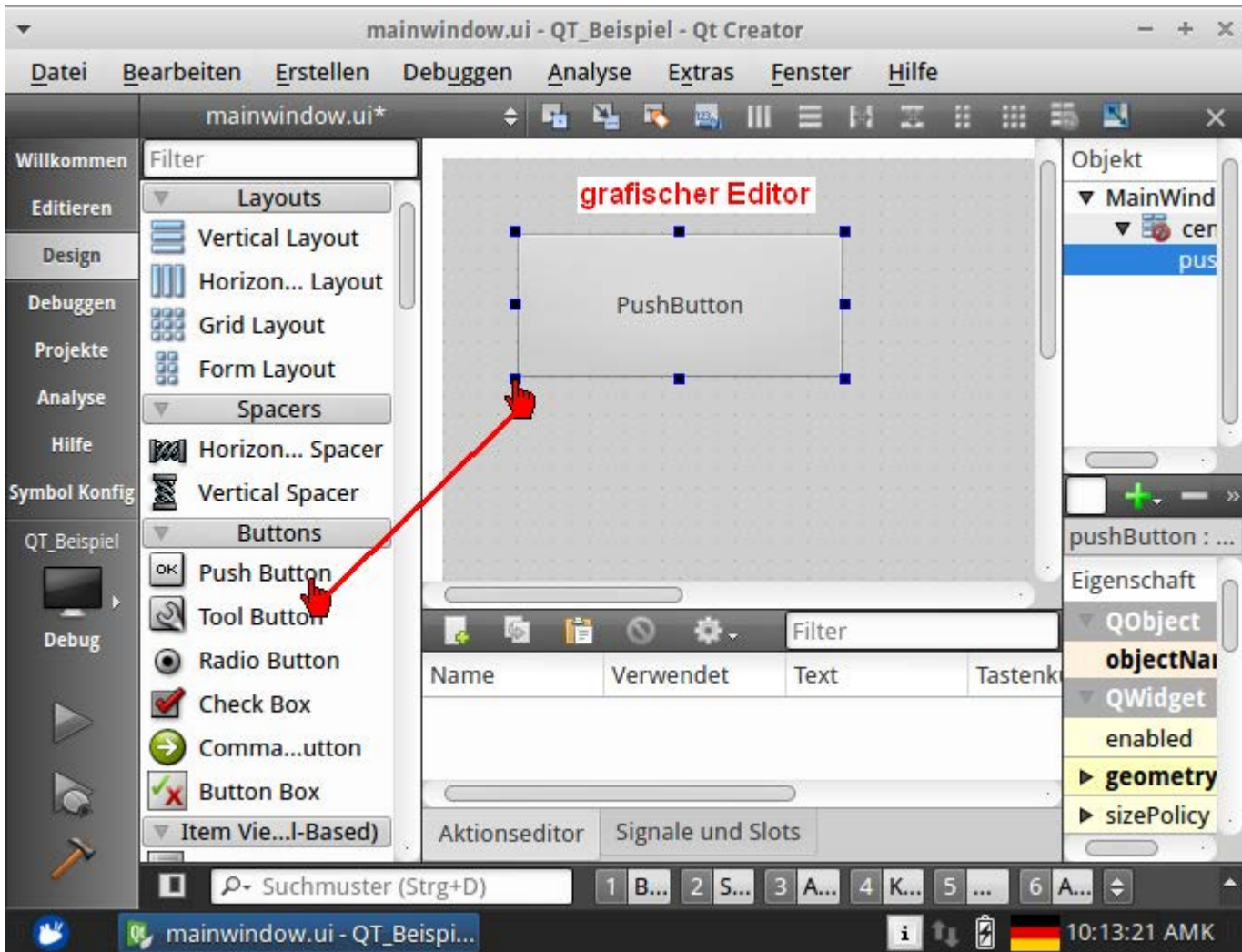
Editieren

Wählen Sie anschließend **'mainwindow.ui'** unter **'Formulardateien'** an.

Schalten Sie (wenn erforderlich) in den Designmodus durch klicken auf die Taste **'Modus umschalten'**.

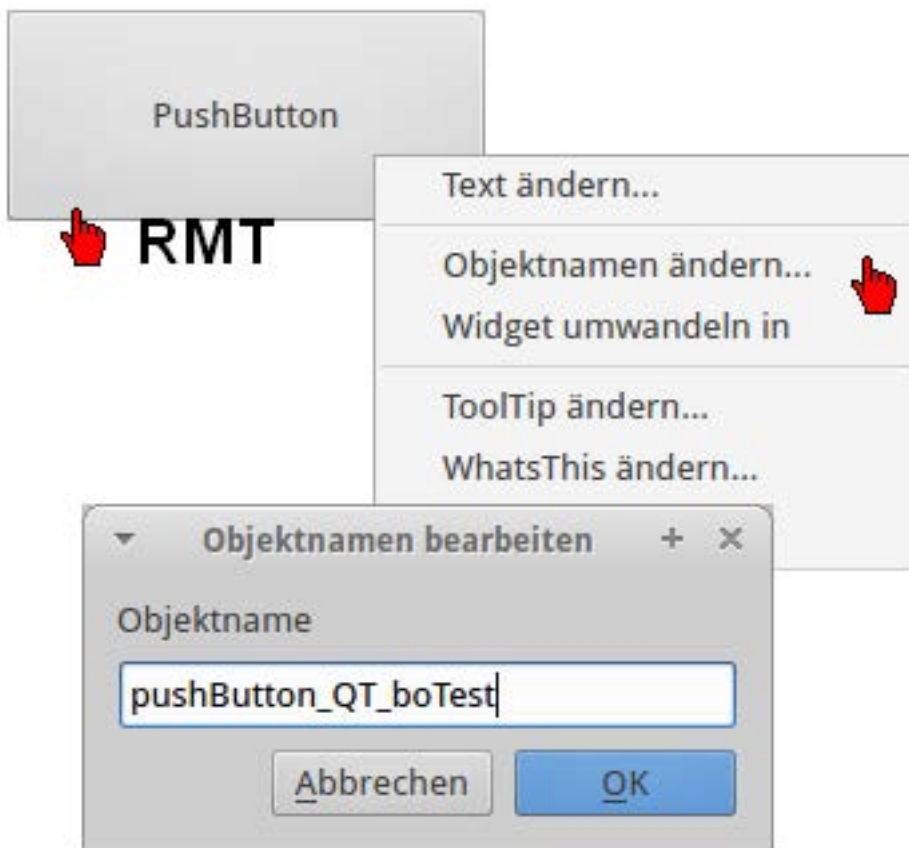


Ziehen Sie einen 'Push Button' per Drag & Drop in den Grafikeditor. Die Objektgröße kann anschließend verändert werden.



Klicken Sie mit der RMT auf das Objekt um einen spezifischen Objektnamen zu vergeben.

Beispiel: pushButton_QT_boTest

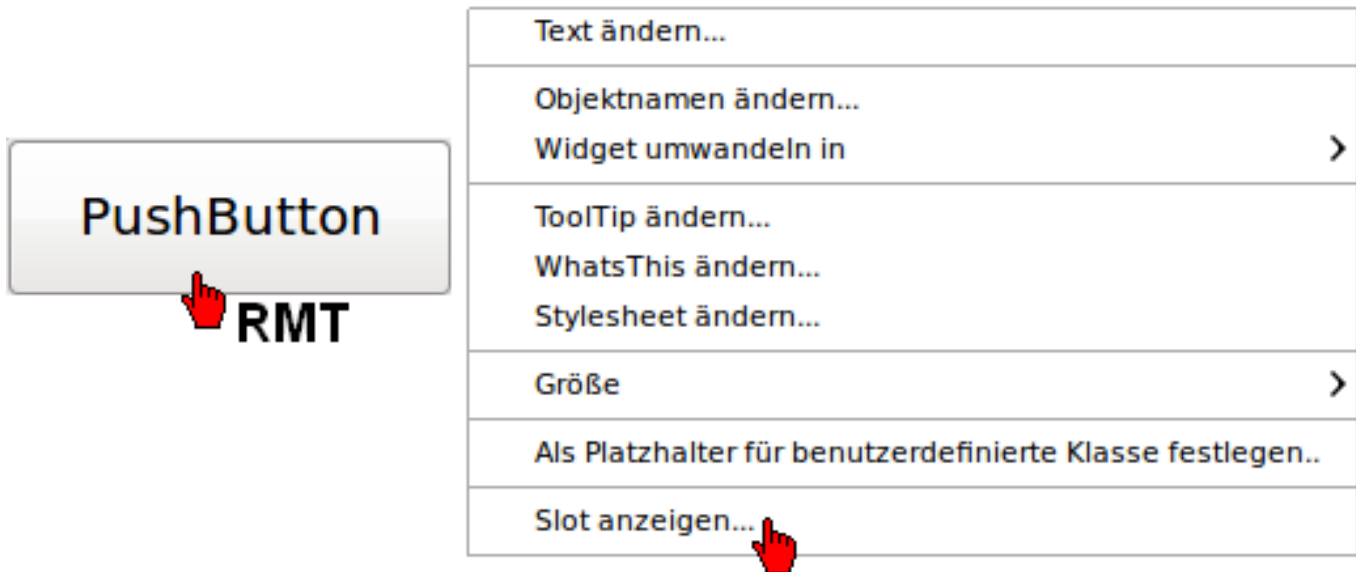


Aktivieren Sie die Button Eigenschaft '**checkable**' für ein toggelnden Button.

pushButton : QPushButton	
Eigenschaft	Wert
inputMethodHints	ImhNone
QAbstractButton	
text	PushButton
icon	
iconSize	24 x 24
shortcut	
checkable	☒
checked	☐
autoRepeat	☐
autoExclusive	☐
autoRepeatDelay	300
autoRepeatInterval	100

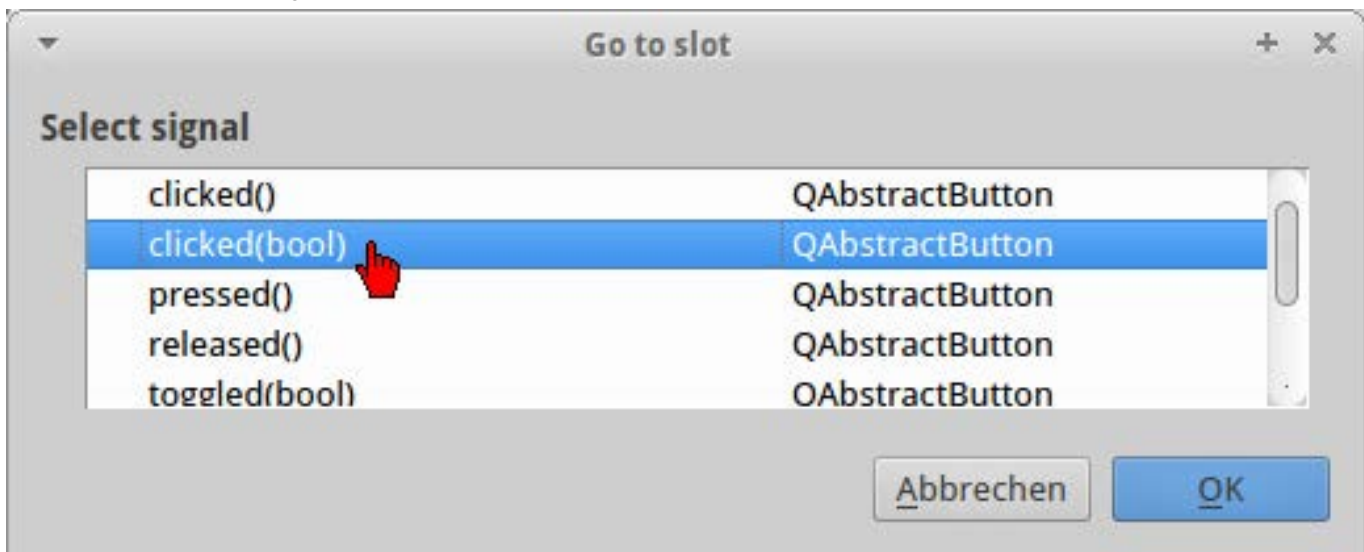
Mit der Funktion 'Slot anzeigen' wird zuerst das 'Button Eigenschaftsfenster' geöffnet. Nach Auswahl der Button Eigenschaft wird automatisch eine Funktion (Methode) im Quellcode erzeugt. Ändert sich der Zustand des Buttons wird die Funktion (Methode) aufgerufen und abgearbeitet.

Klicken Sie mit der RMT auf den Push Button. Wählen Sie anschließend die Funktion '**Slot anzeigen**' aus.

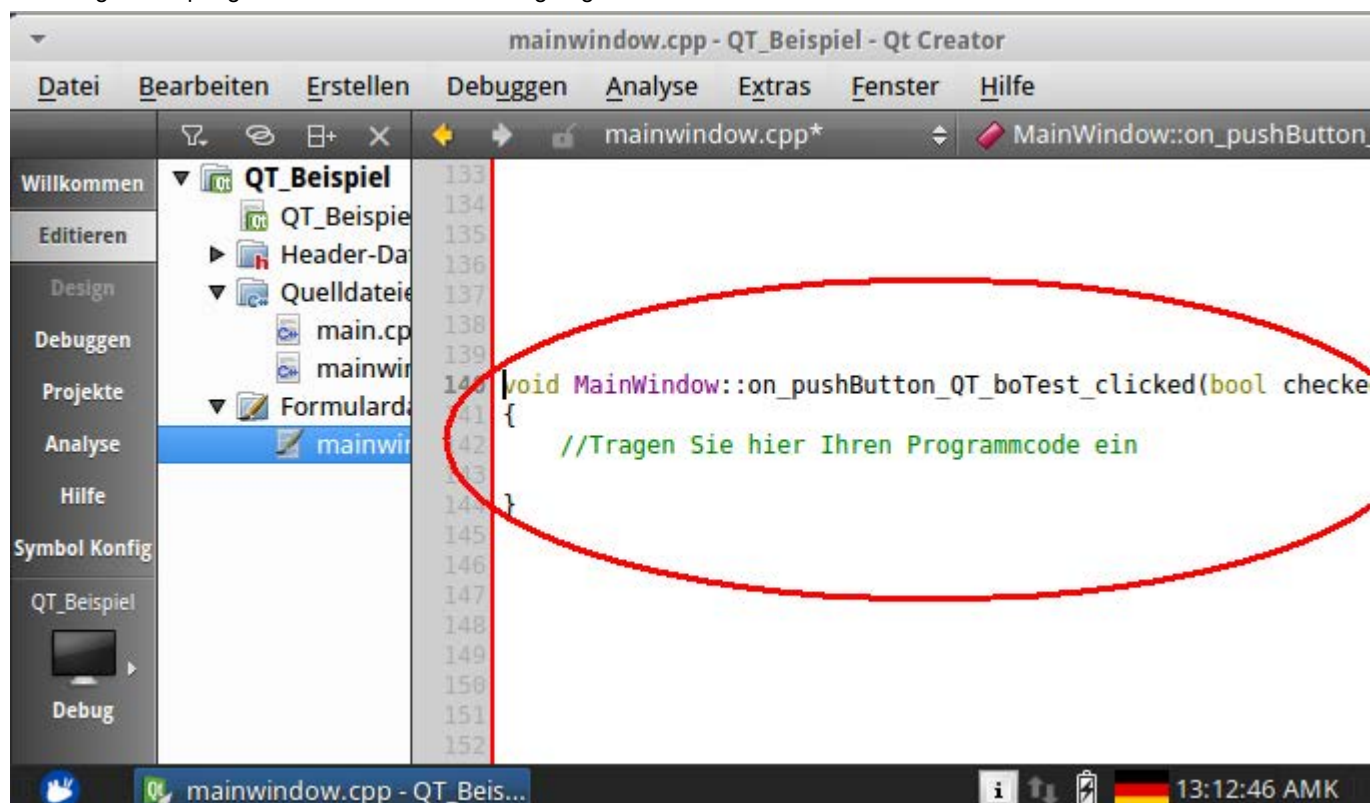


Bei gedrücktem Button (positive Flanke) wird die Variable <QT_boTest>; auf 'true' gesetzt. Bei negativer Flanke wird die Variable <QT_boTest> auf 'false' zurückgesetzt.

Verwenden Sie dazu die Eigenschaft '**clicked(bool)**'.



Das Programm springt automatisch an die neu eingefügte Funktion.



Verwenden Sie den `m_plcHandler` um mit der Visu Variable `QT_boTest` auf `boTest` der PLC Steuerung zuzugreifen.

```
void MainWindow::on_pushButton_QT_boTest_clicked(bool checked)
{
    m_plcHandler.writePlcValue(QT_boTest, checked);
}
```

'`m_plcHandler`' ist das Objekt, mit dem auf die PLC Steuerung zugegriffen werden kann. Hier wird die Funktion '`writePlcValue` (Variable, Wert)' des '`m_plcHandler`'-Objektes benutzt um die Variable in der PLC zu schreiben. Der Variablenname '`QT_boTest`' ist hierbei der Variablenname, der im 'AMK Symbol Config' Dialog angelegt und mit `boTest` der PLC Steuerung verknüpft wurde. '`checked`' ist der Zustand des Buttons.

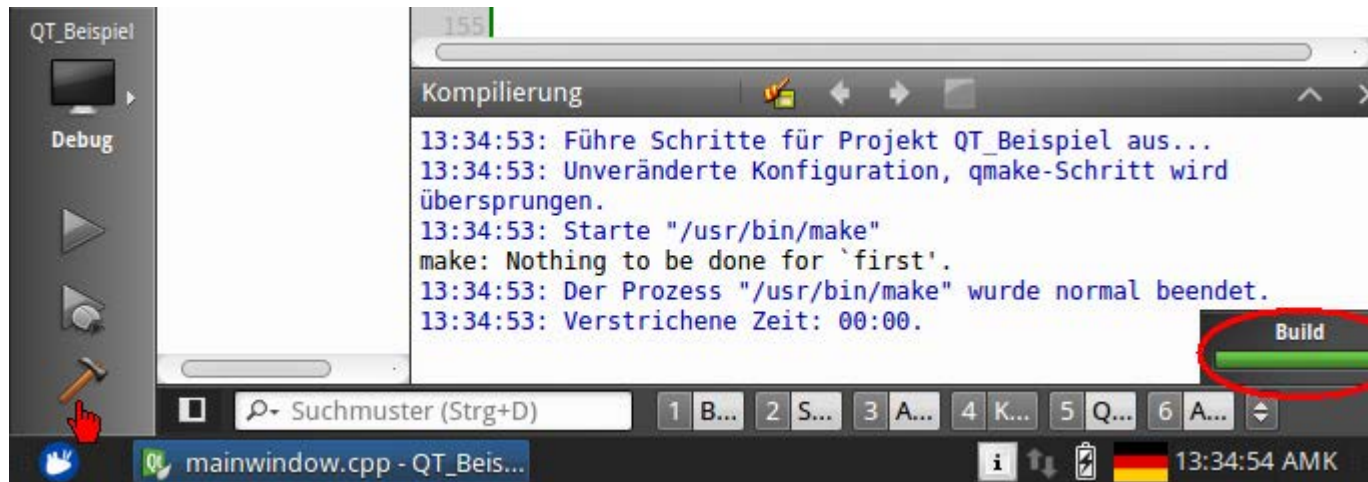
5.5.4 Qt Visualisierung übersetzen

In diesem Kapitel wird beschrieben, wie Sie den Sie die erstellte Grafik und den Programmcode kompilieren.

Sie können dazu

- den Hammer 'Alles erstellen'
- das Menü 'Erstellen' 'Alles erstellen'
- die 'F7' Taste
- oder im Kontextmenü des Projektbaumes 'Projekt' 'Projektname erstellen' auswählen.

Die Anzeige 'Erstellen' gibt den aktuellen Status wieder.



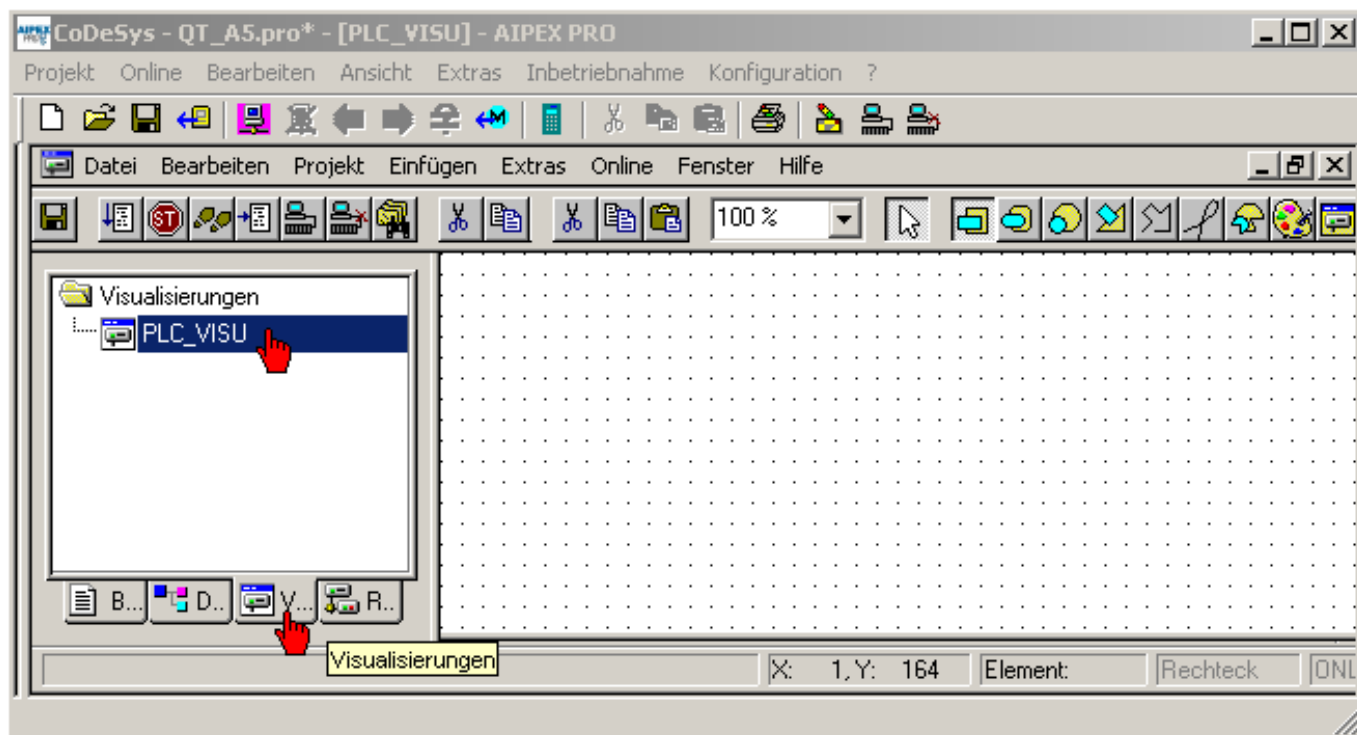
Der 'Qt Creator' erstellt eine Datei mit dem Namen 'PLCvisu'. Diese wird im Verzeichnis 'C:\AMK\src\...' gespeichert. Die Datei PLCvisu wird anschließend in das CODESYS Projekt integriert.

5.6 CODESYS fertigstellen

5.6.1 Qt Visualisierung unter CODESYS V2 integrieren

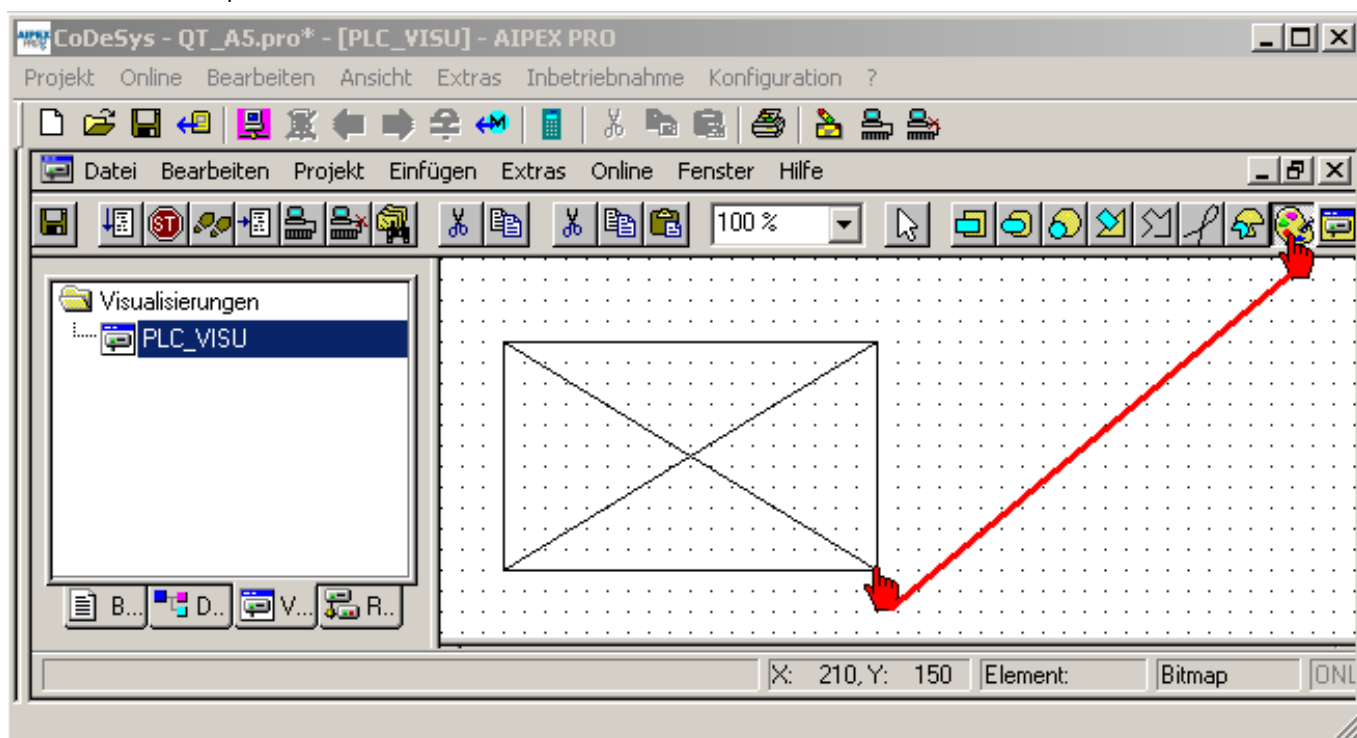
Das Kapitel beschreibt, wie Sie die vom 'Qt Creator' erstellte Datei 'PLCvisu' in die CODESYS Visualisierung integrieren.

Aktivieren Sie den Reiter 'Visualisierungen', danach können Sie die Seite PLC-VISU' öffnen.



Integrieren Sie per Drag & Drop ein Bitmap-Feld in die CODESYS Visualisierung.

Die Größe des Bitmap-Feldes ist dabei unbedeutend.

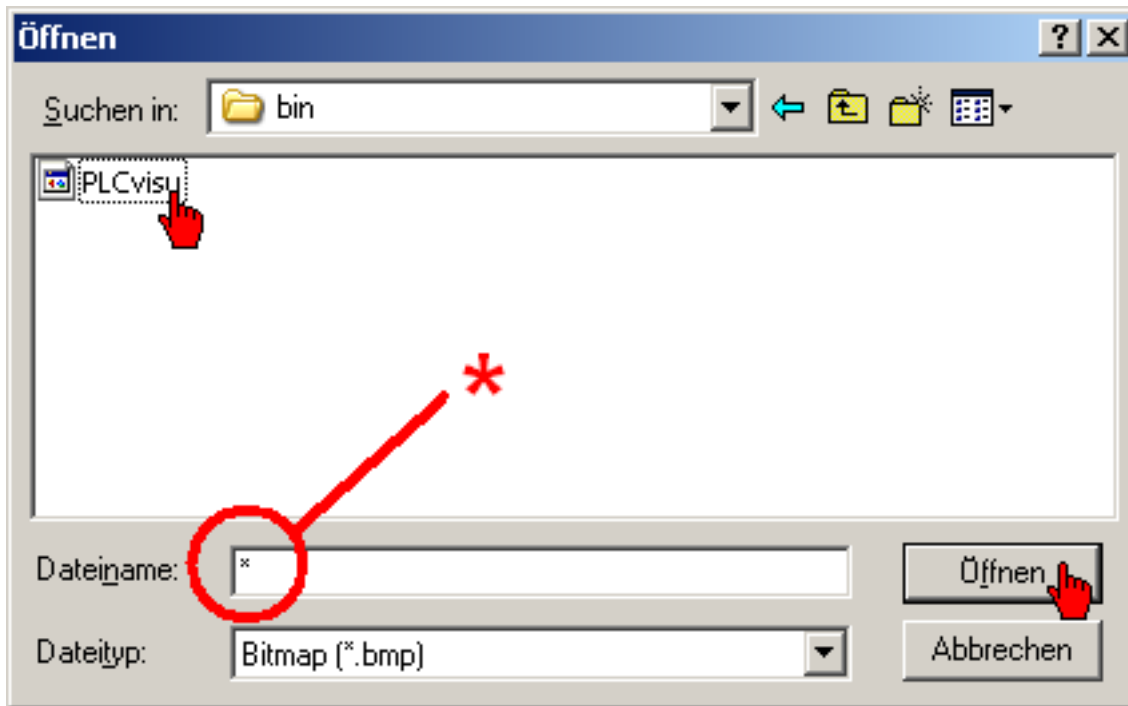


Öffnen Sie den Pfad: 'C:\AMK\src\QT_Beispiel\QT_Beispiel\bin'.



Die Qt Datei 'PLCvisu' wird erst sichtbar nachdem Sie in das Feld 'Dateiname' einen '*' eingetragen und die Taste **'Enter'** gedrückt haben.

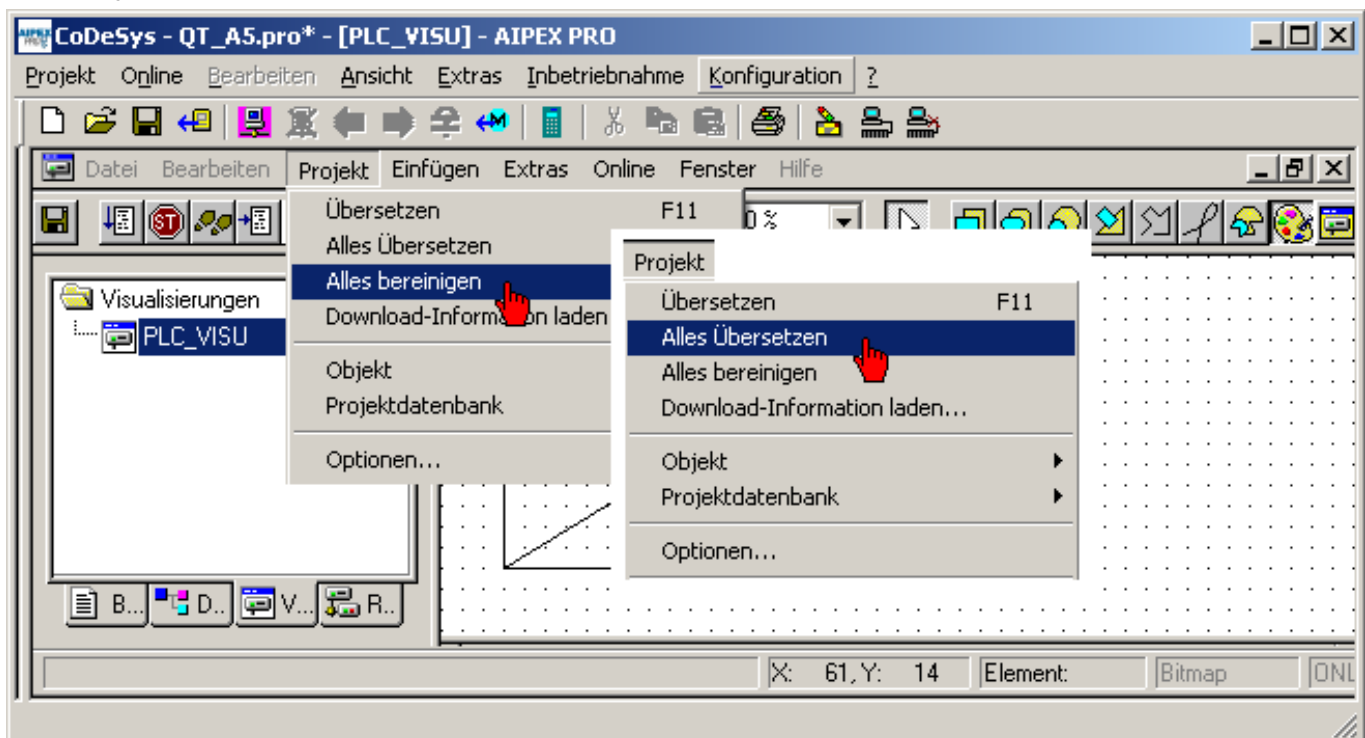
Öffnen Sie die Datei 'PLCvisu'.



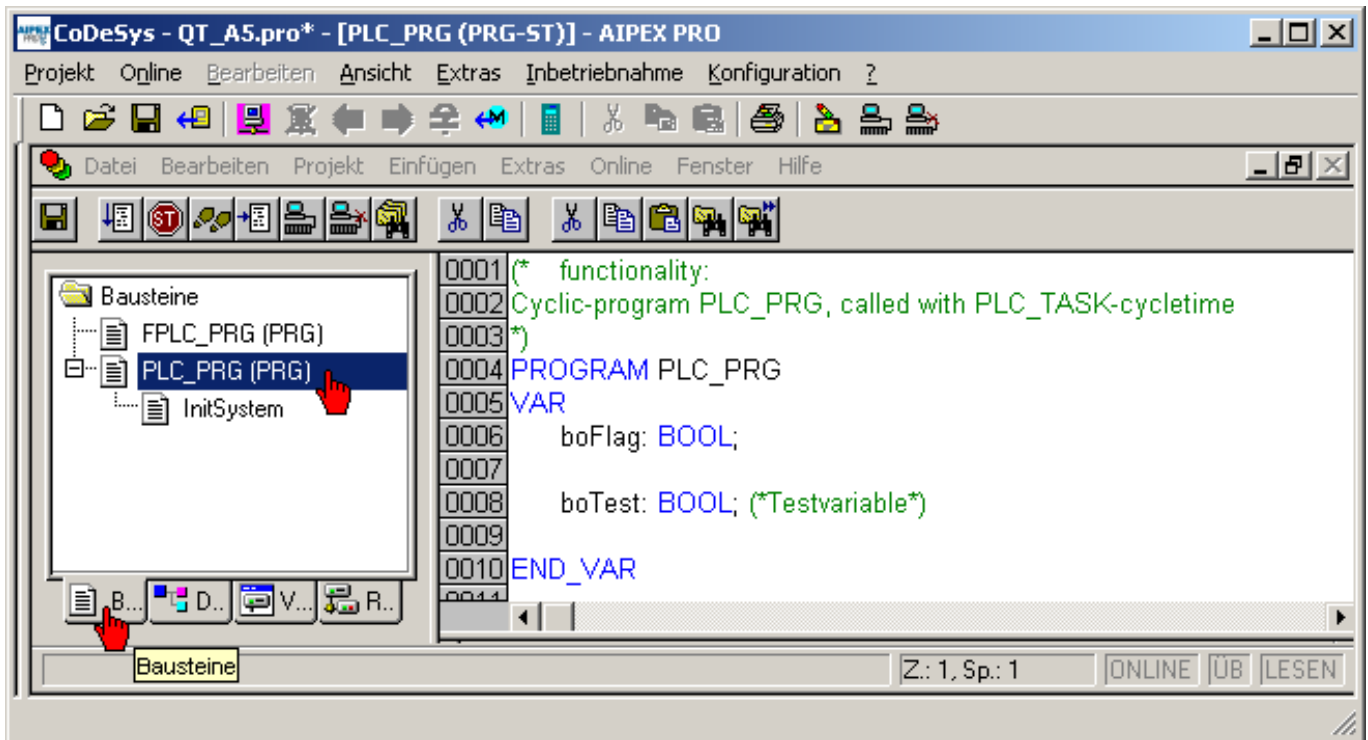
5.6.1.1 CODESYS V2 Projekt Fertigstellen und Testen

In diesem Kapitel wird beschrieben, wie Sie das CODESYS Projekt Übersetzen, an die PLC Steuerung Übertragen und die Kommunikation zwischen der Qt Visualisierung und dem PLC Programm testen.

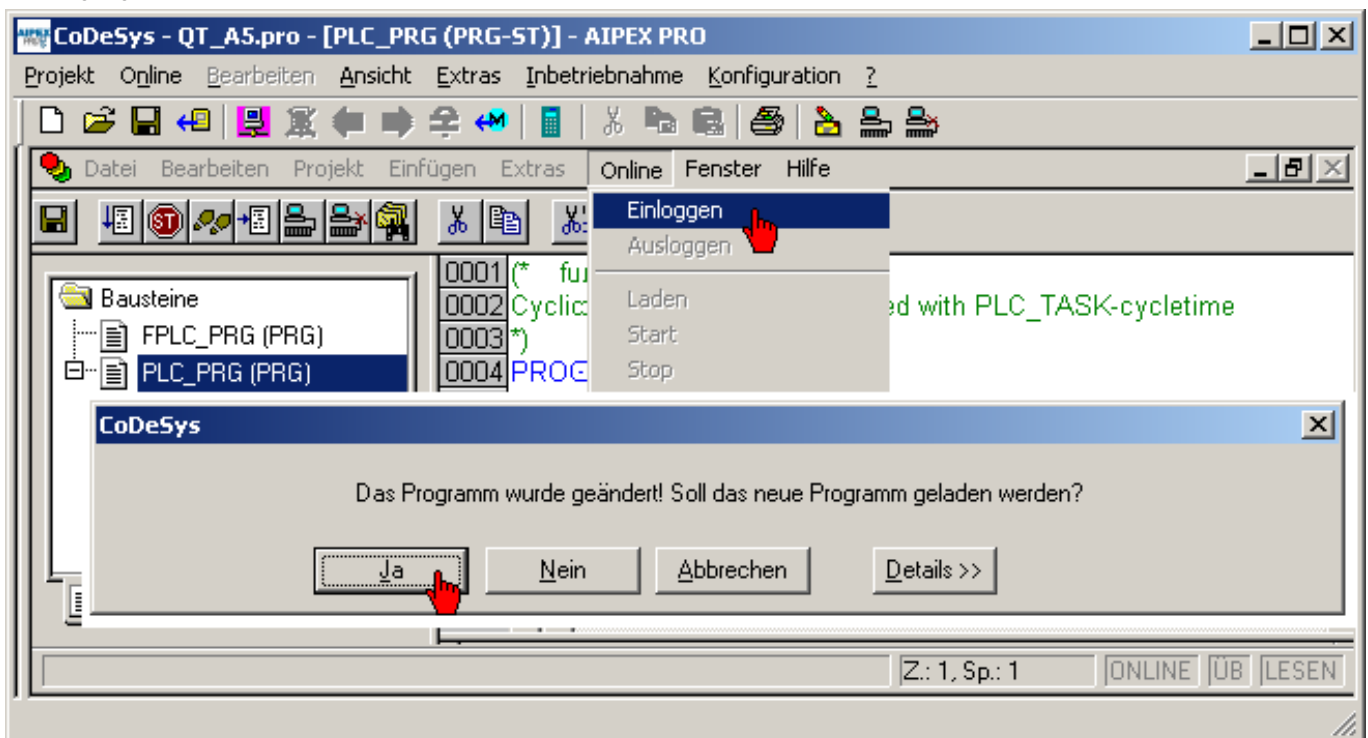
Bereinigen Sie zuerst das Projekt. Menü **'Projekt' 'Alles Bereinigen'**. Anschließend wird das komplette Projekt neu übersetzt. Menü **'Projekt' 'Alles Übersetzen'**.



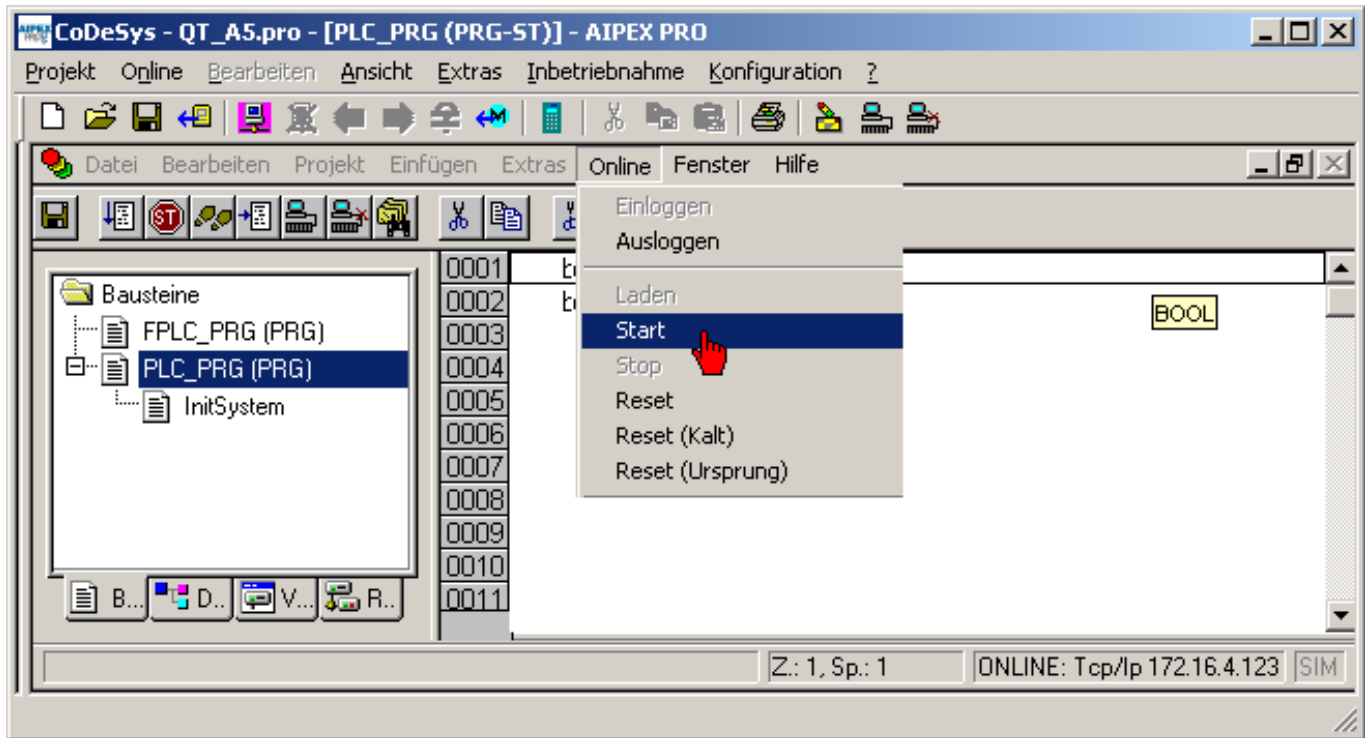
Wechseln Sie auf den Programmbaustein 'PLC_PRG' im Reiter 'Bausteine'.



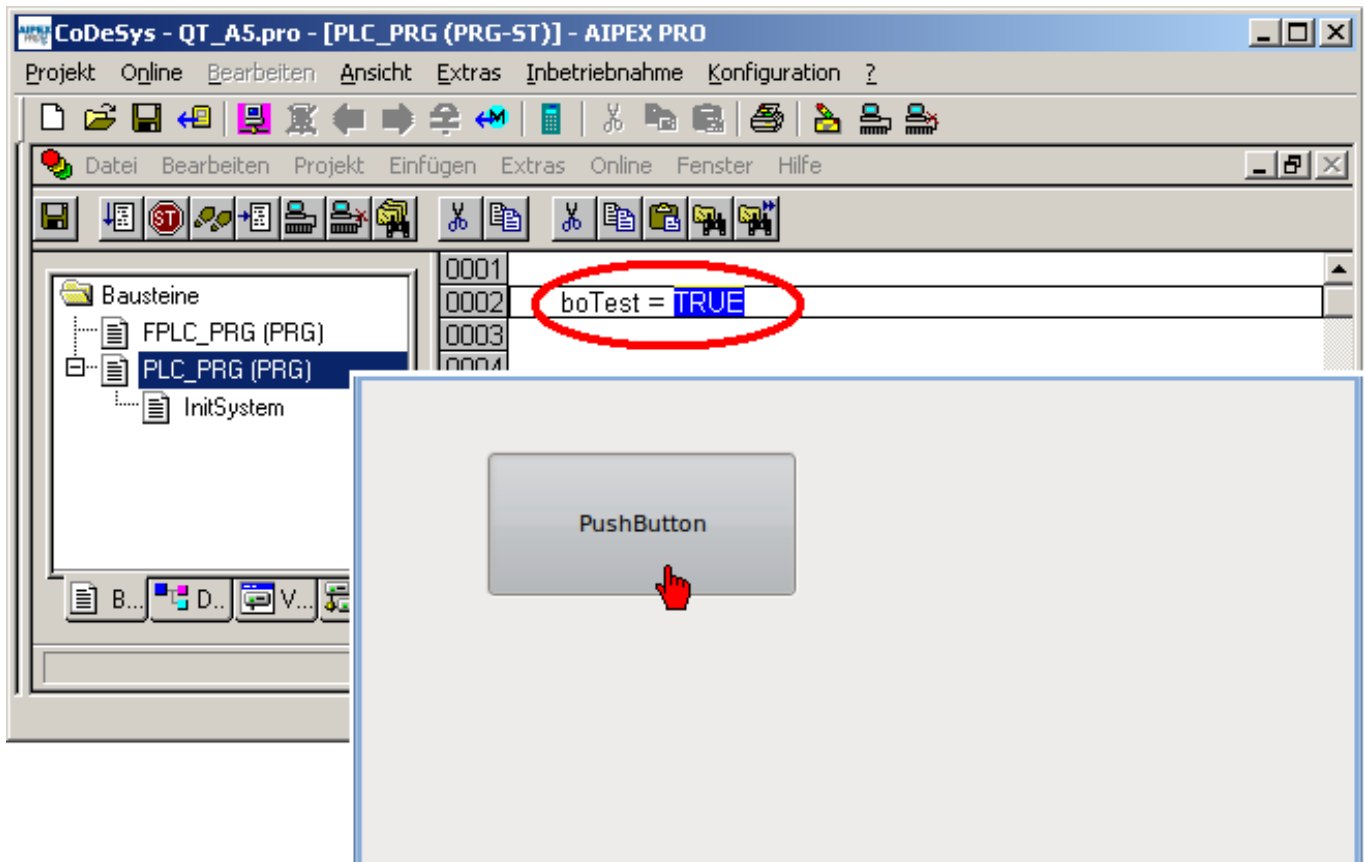
Loggen Sie sich über 'Online' 'Einloggen' an der Steuerung ein. Übertragen Sie anschließend den Programmcode durch Bestätigung mit der 'Ja' Taste.



Starten Sie das PLC Programm im Menü **'Online' 'Start'**. Erstellen Sie ein Bootprojekt, sofern Sie das PLC Programm remanent speichern möchten. Menü **'Online' 'Bootprojekt erzeugen'**.



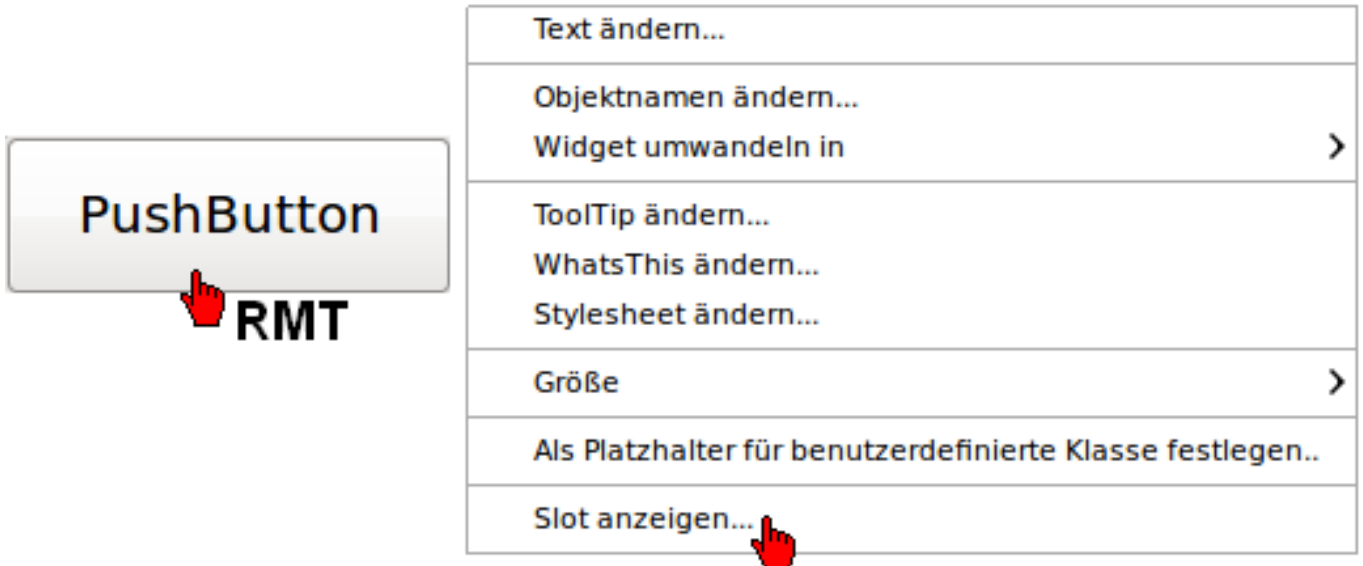
Testen Sie Ihr Programm. Der Status des Toggel Buttons wird immer beim loslassen der Taste übernommen.



5.7 Weitere Beispiele

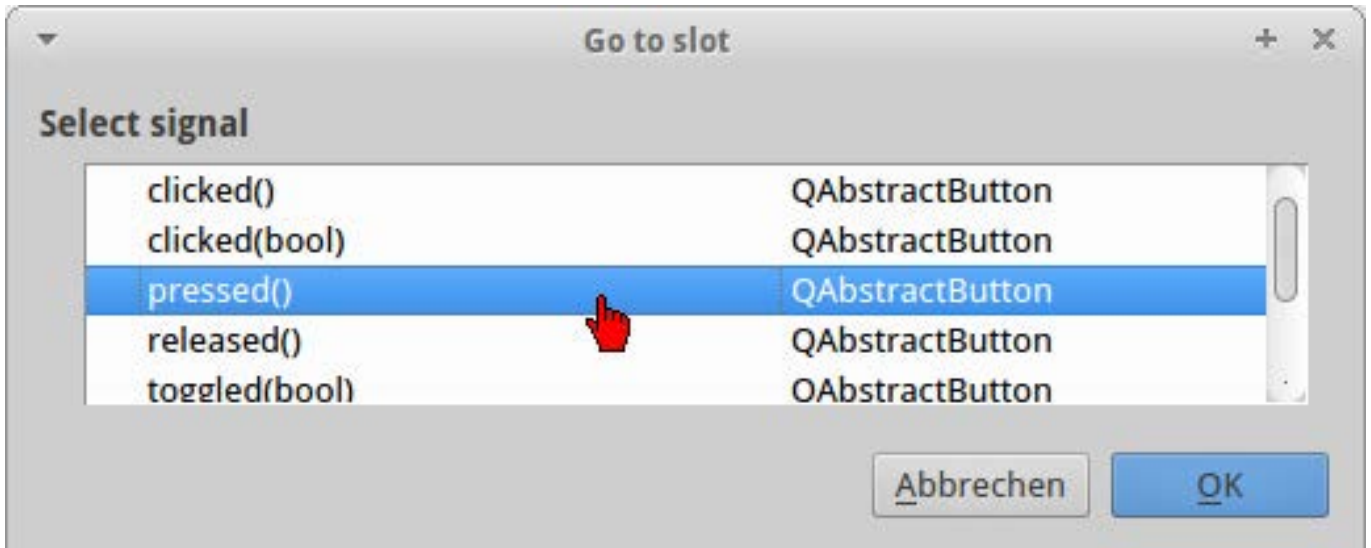
5.7.1 Button (tastend)

Beispiel: Button tastend um z.B. das Signal **'Fehler Löschen'** auszulösen.



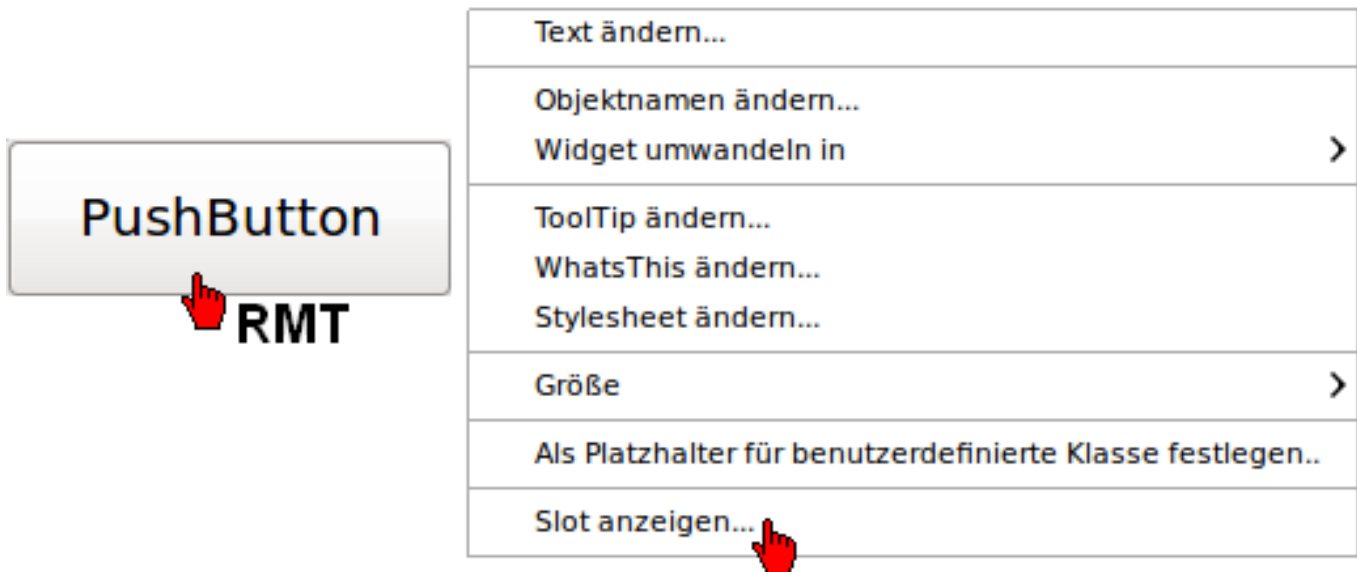
Bei gedrücktem Button (positive Flanke) wird die Variable <boFL_Drive1> auf **'true'** gesetzt.

Verwenden Sie dazu die Eigenschaft **'pressed()'**.

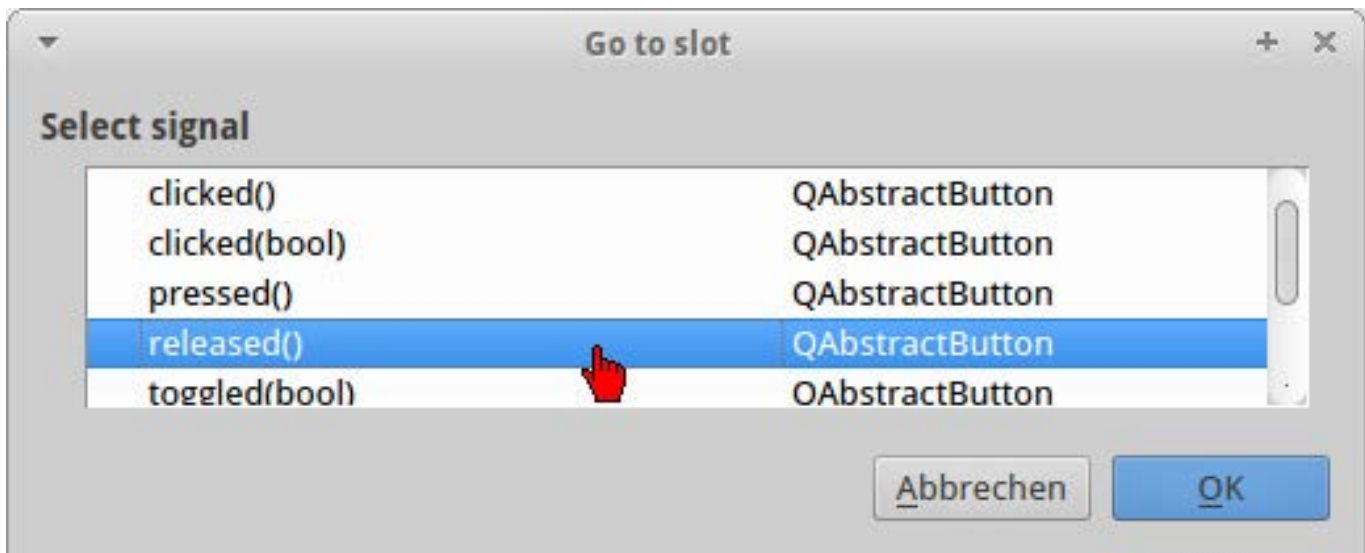


```
void MainWindow::on_pushButton_pressed()
{
    m_plcHandler.writePlcValue(boFL_Drive1, true);
}
```

Wechsel Sie zurück in den Designer um einen zweiten Slot einzufügen.



Beim loslassen (negativer Flanke) wird die Variable <boFL_Drive1> auf **'false'** gesetzt.
Verwenden Sie dazu die Eigenschaft **'released()'**.



```
void MainWindow::on_pushButton_released()
{
    m_plcHandler.writePlcValue(boFL_Drive1, false);
}
```

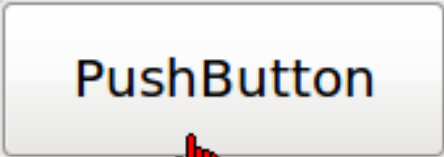
5.7.2 Button (toggleInd)

Beispiel: Toggle-Button um z.B. das Signal '**Reglerfreigabe**' zu setzen/rücksetzen.

Bedeutung toggeln: Button wird einmalig betätigt. Der Zustand bleibt bis zur nächsten Betätigung erhalten.

Aktivieren Sie die Button Eigenschaft '**checkable**' für ein togglenden Button.

pushButton : QPushButton	
Eigenschaft	Wert
inputMethodHints	ImhNone
QAbstractButton	
text	PushButton
icon	
iconSize	24 x 24
shortcut	
checkable	☒
checked	☐
autoRepeat	☐
autoExclusive	☐
autoRepeatDelay	300
autoRepeatInterval	100

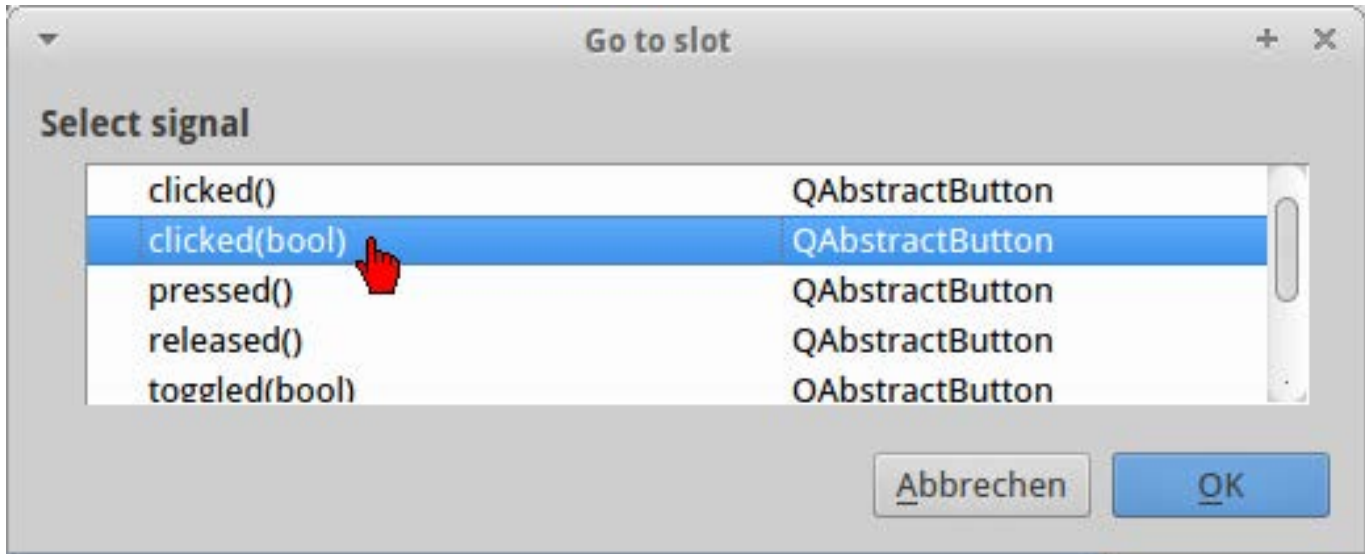


RMT

Text ändern...
Objektnamen ändern...
Widget umwandeln in >
ToolTip ändern...
WhatsThis ändern...
Stylesheet ändern...
Größe >
Als Platzhalter für benutzerdefinierte Klasse festlegen..
Slot anzeigen... 

Bei gedrücktem Button (positive Flanke) wird die Variable <boRF>; auf 'true' gesetzt. Bei negativer Flanke wird die Variable <boRF> auf **'false'** zurückgesetzt.

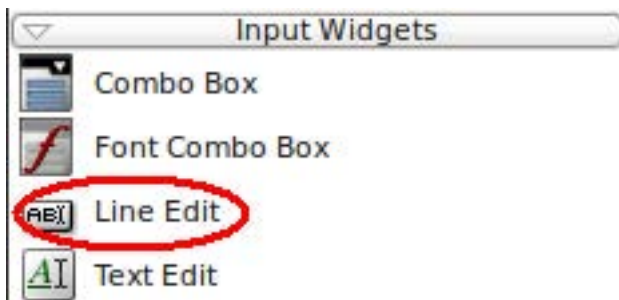
Verwenden Sie dazu die Eigenschaft **'clicked(bool)'**.



```
void MainWindow::on_pushButton_clicked(bool checked)
{
    m_plcHandler.writePlcValue(boRF, checked);
}
```

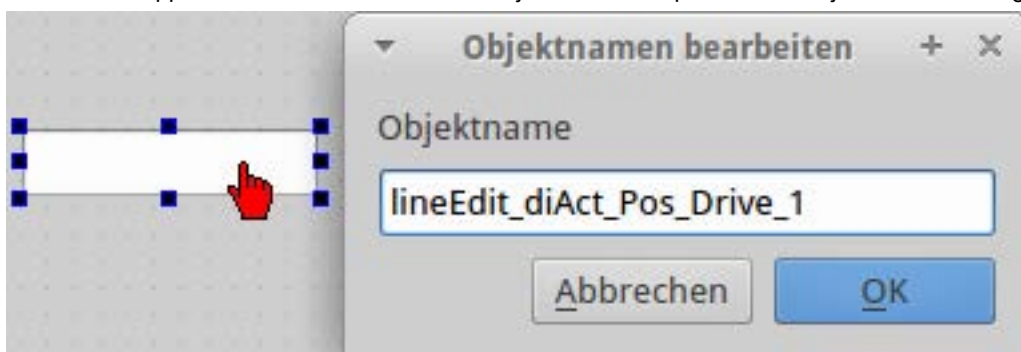
5.7.3 Istwerte anzeigen (Line Edit)

Mit dem Objekt **'Line Edit'** könne Sie Istwerte wie z.B. Position, Drehzahl und Moment visualisieren.



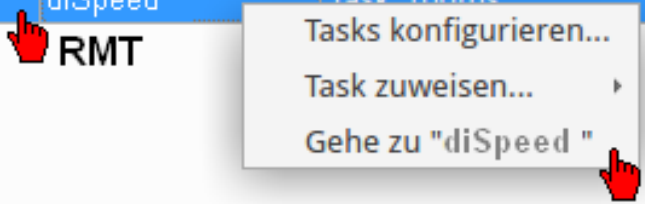
Klicken Sie mit der LMT auf das Objekt. Mit gedrückter LMT können Sie das Objekt im Editor platzieren. Anschließend können Sie Form anpassen.

Klicken Sie doppelt bzw. mit der RMT auf das Objekt um einen spezifischen Objektnamen zu vergeben.



Klicken Sie mit der RMT auf die Variable, die Sie mit dem Visualisierungsobjekt verbinden möchten.
Anschließend können Sie über das Dialogfeld (Gehe zu "****") zum Programmcode wechseln.

PS Variablen-Nam ▲	su Variablen-Nan	Task
PLC_PRG.boQUE	boQUE	Keine Liste (nur Funktionaler Aufruf)
PLC_PRG.boRF	boRF	Keine Liste (nur Funktionaler Aufruf)
PLC_PRG.boTest	boTest	Keine Liste (nur Funktionaler Aufruf)
PLC_PRG.boUE	boUE	Keine Liste (nur Funktionaler Aufruf)
PLC_PRG.diSpeed	diSpeed	Task 100ms



Ergänzen Sie den automatisch generierten Programmcode.

Der Wert der Variable diAct_Pos_Drive1 wird im Line Edit Feld lineEdit_diAct_Pos_Drive1 angezeigt.

```
void MainWindow::on_diAct_Pos_Drive1_changed(const QVariant &value)
{
    //Add your code below
    ui.lineEdit_diAct_Pos_Drive1->setText(QString::number(value.toInt()));
}
```

Beispiel: Drehzahlwert, Skalierung und 2 Nachkommastellen werden berücksichtigt.

```
void MainWindow::on_di_ACT_Velocity_changed(const QVariant &value)
{
    //Add your code below
    double velocity = value.toDouble()/10000.0;
    ui.diActVelocity->setText(QString::number(velocity, 'f',2)); // 2 Nachkommastellen
}
```

5.7.4 Istwerte anzeigen (ManoMeter)

Mit dem Objekt 'ManoMeter' können Sie Istwerte wie z.B. Drehzahl und Moment visualisieren.

Klicken Sie mit der LMT auf das Objekt. Mit gedrückter LMT können Sie das Objekt im Editor platzieren. Anschließend können Sie Form anpassen.

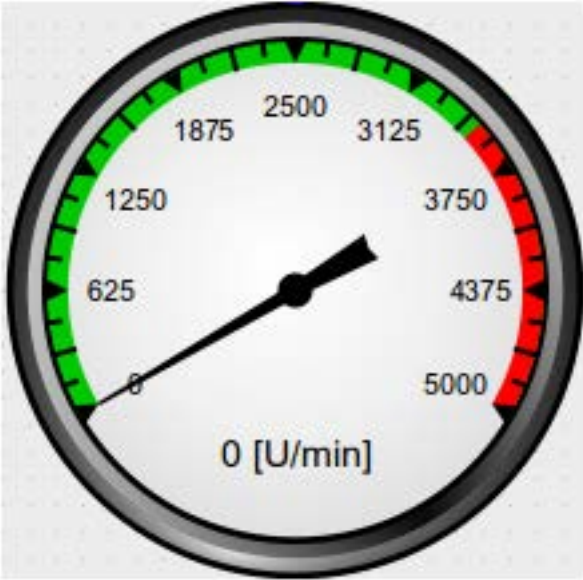
Klicken Sie doppelt bzw. mit der RMT auf das Objekt um einen spezifischen Objektnamen zu vergeben. (im Beispiel 'Drehzahl')

Im Eigenschaftsfenster können Sie Ihr Manometer anpassen.

▼ Analog Widgets

Qt WallClock

Qt ManoMeter



Drehzahl : ManoMeter	
Eigenschaft	Wert
▼ AbstractMeter	
minimum	0.000000
maximum	5000.000000
value	0.000000
nominal	0.000000
critical	3500.000000
▷ prefix	
▷ suffix	[U/min]
▷ valueFont	A [Arial Unicode MS, 16]
valueOffset	-100.000000
▷ digitFont	A [Arial Unicode MS, 12]
digitOffset	105.000000

Klicken Sie mit der RMT auf die Variable, die Sie mit dem Visualisierungsobjekt verbinden möchten.

Anschließend können Sie über das Dialogfeld (Gehe zu "****") zum Programmcode wechseln.

PS Variablen-Nam ▲	su Variablen-Nan	Task
PLC_PRG.boQUE	boQUE	Keine Liste (nur Funktionaler Aufruf)
PLC_PRG.boRF	boRF	Keine Liste (nur Funktionaler Aufruf)
PLC_PRG.boTest	boTest	Keine Liste (nur Funktionaler Aufruf)
PLC_PRG.boUE	boUE	Keine Liste (nur Funktionaler Aufruf)
PLC_PRG.diSpeed	diSpeed	Task 100ms

 **RMT**

Tasks konfigurieren...

Task zuweisen...

Gehe zu "diSpeed "

Ergänzen Sie den automatisch generierten Programmcode.

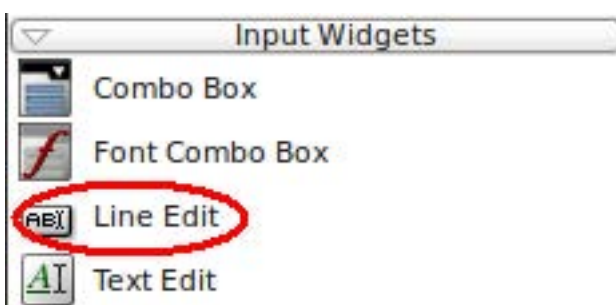
```
void MainWindow::on_diSpeed_Drive1_changed(const QVariant &value)
{
    //Add your code below
    ui.Drehzahl->setValue(value.toInt());
}
```

5.7.5 Sollwerte vorgeben (über Numpad)

Mit dem Objekt '**Line Edit**' könne Sie z.B. Sollwerte vorgeben.

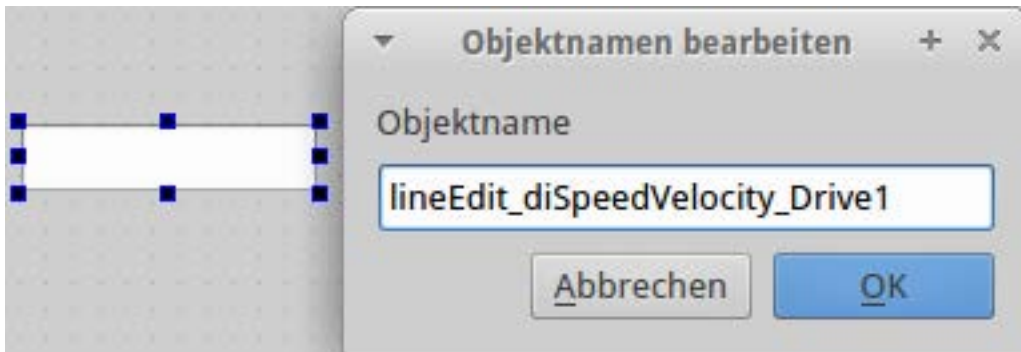
In diesem Beispiel wird das Objekt '**Line Edit**' mit einem Numpad verknüpft.

Der Variablenname lautet diSpeedVelocity_Drive1.



Klicken Sie mit der LMT auf das Objekt. Mit gedrückter LMT können Sie das Objekt im Editor platzieren. Anschließend können Sie Form anpassen.

Klicken Sie doppelt bzw. mit der RMT auf das Objekt um einen spezifischen Objektnamen zu vergeben.



Um ein Numpad darzustellen, benötigen Sie das Headerfile numkeypaddlg.h.

Programmcode '**Quelldateien**' '**mainwindow.cpp**'

```
#include <numkeypaddlg.h> // Headerfile für Numpad
```

Erweitern Sie den Konstruktor um folgenden Programmcode:

```
ui.lineEdit_diSpeedVelocity_Drive1->installEventFilter(this);
```

```
MainWindow::MainWindow(QWidget *parent):
    QMainWindow(parent),
    m_plcHandler()
{
    ui.setupUi(this);
    initPLC();

    ui.lineEdit_diSpeedVelocity_Drive1->installEventFilter(this);
    // Sollwertvorgabe über Numpad
}
```

//Eventfilter, here you can set the key-/numpad dialog

```
bool MainWindow::eventFilter(QObject *object, QEvent *event)
```

```
{
    Q_UNUSED(object);
    if (event->type() == QEvent::MouseButtonPress)
    {
        if (object==ui.lineEdit_diSpeedVelocity_Drive1)
        {
            NumKeypadDlg dlg(tr("Set Velocity"),"-2000", "2000",ui.lineEdit_diSpeedVelocity_Drive1->text());

            connect(&dlg, SIGNAL(sendValue(QString)), ui.lineEdit_diSpeedVelocity_Drive1, SLOT(setText
            (QString)));

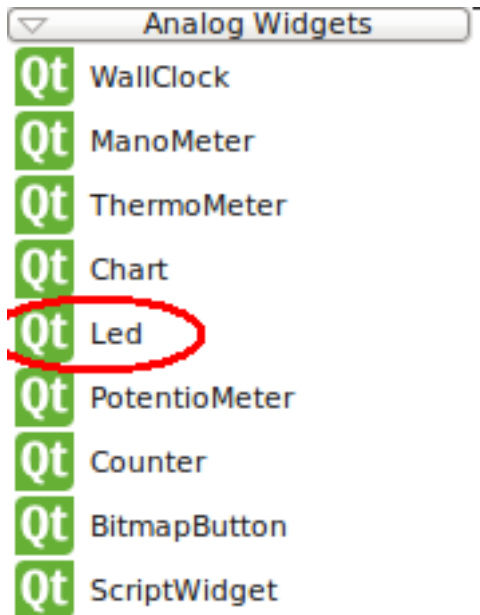
            dlg.exec();

            m_plcHandler.writePlcValue(diSpeedVelocity_Drive1, ui.lineEdit_diSpeedVelocity_Drive1->text
            ().toInt());
        }
    }
    return false;
}
```

Überschrift Numpad 'tr' wegen eventueller Sprachumschaltung

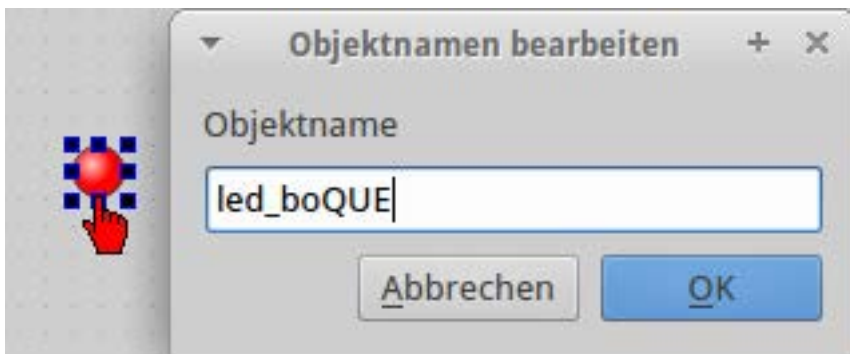
5.7.6 Statussignale (LED)

Mit dem Objekt 'LED' können Sie Statussignale wie z.B. QUE, QRF, SBM... visualisieren.



Klicken Sie mit der LMT auf das Objekt. Mit gedrückter LMT können Sie das Objekt im Editor platzieren. Anschließend können Sie Form anpassen.

Klicken Sie doppelt bzw. mit der RMT auf das Objekt um einen spezifischen Objektname zu vergeben.



Klicken Sie mit der RMT auf die Variable, die Sie mit dem Visualisierungsobjekt verbinden möchten.
Anschließend können Sie über das Dialogfeld (Gehe zu "****") zum Programmcode wechseln.

AMK
Konfiguration Kommunikation Qt <--> SPS

Symbol Datei: öffnen

PLC_PRG.boQUE
 PLC_PRG.boRF
 PLC_PRG.boTest
 PLC_PRG.boUE

▶

▶▶

◀◀

◀

	SPS Variablen-Name	Visu Variablen-Name	Task
0	PLC_PRG.boTest	boTest	Keine Liste (nur Funtionaler Au
1	PLC_PRG.boUE	boUE	Keine Liste (nur Funtionaler Au
2	PLC_PRG.boRF	boRF	Keine Liste (nur Funtionaler Au
3	PLC_PRG.boQUE	boQUE	100ms Task

RMT

Tasks konfigurieren...
 Task zuweisen... >
Gehe zu "boQUE"

Übernehmen und Code erzeugen

Ergänzen Sie den automatisch generierten Programmcode.

Im Beispiel wird bei boQUE = TRUE die LED grün, bei boQUE = FALSE die LED rot.

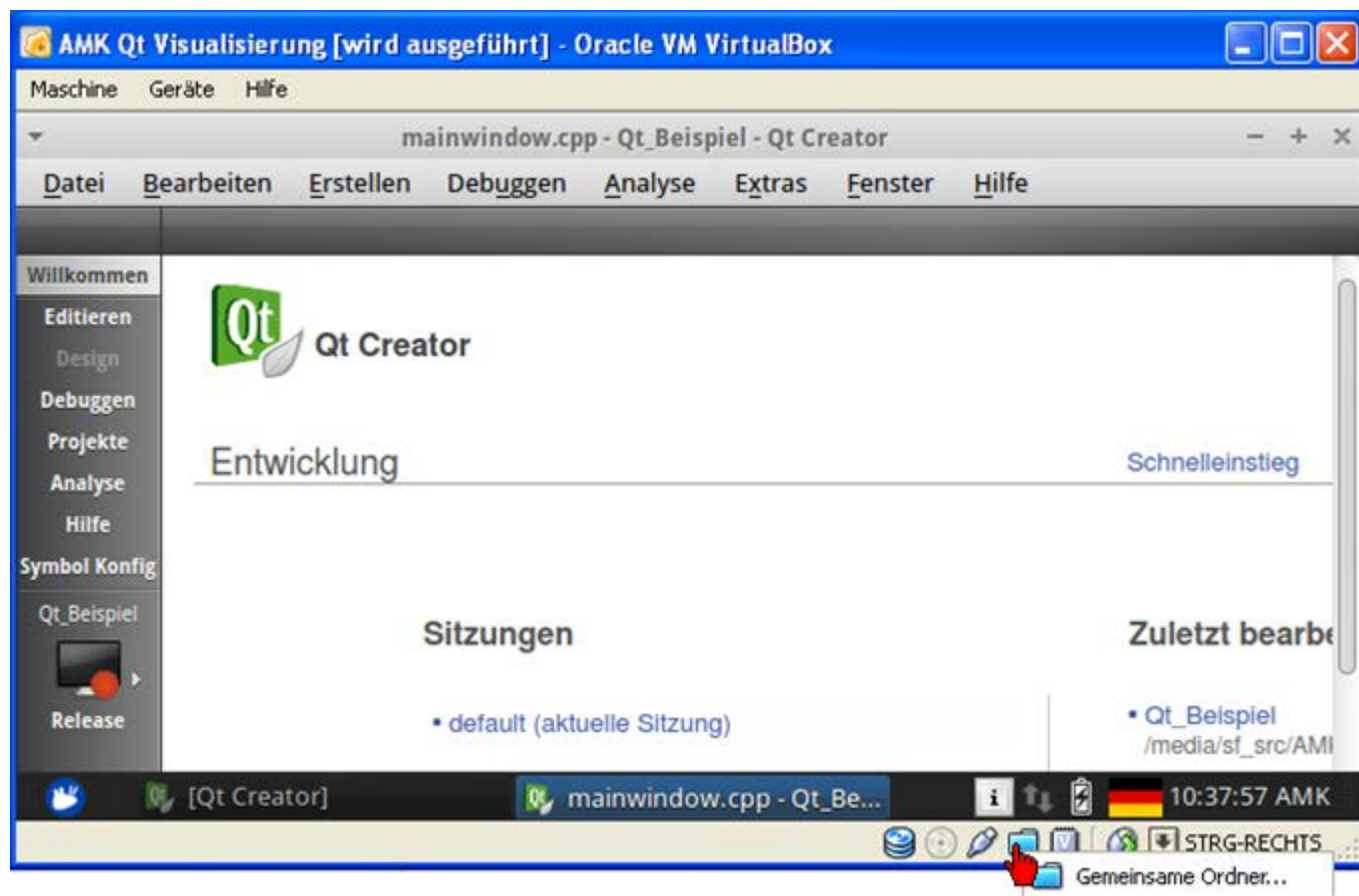
```
void MainWindow::on_boQUE_changed (const QVariant &value)
{
    //Add your code below
    if (value.toBool())
        ui.led_boQUE ->setColor (Qt::green);
    else
        ui.led_boQUE ->setColor(Qt::red);
}
```

5.8 Tipps

5.8.1 Anwenderspezifischer Ordner

Standardmäßig kann die 'Virtuelle Maschine' nur auf den Ordner 'C:\AMK\src\' zugreifen. Das Beispiel zeigt wie Sie anwenderspezifische Ordner verwenden können.

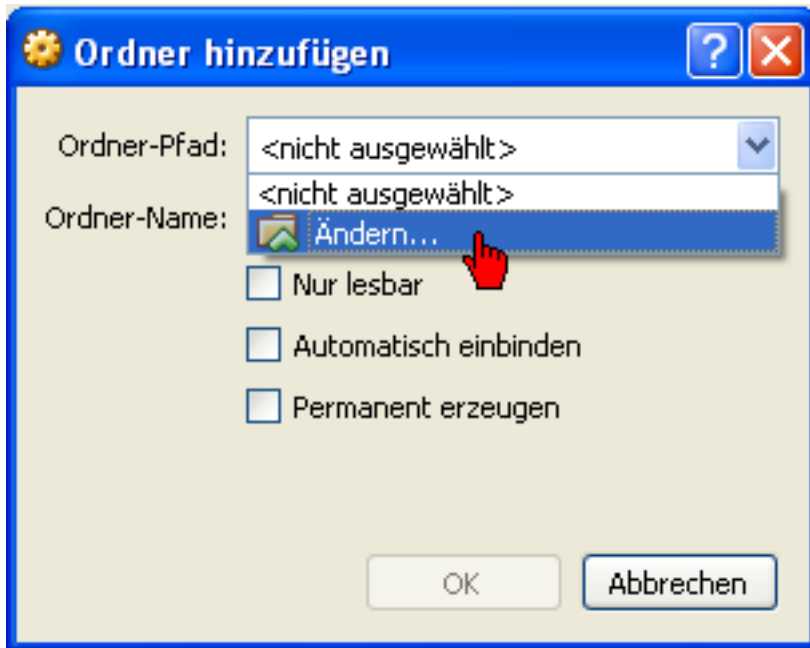
Öffnen Sie in der Qt Task Leiste mit der RMT das Menü  'Gemeinsame Ordner'. Klicken Sie anschließend auf '**Gemeinsame Ordner**'.



Klicken Sie auf  'Gemeinsamen Ordner' hinzufügen.



Stellen Sie das Verzeichnis ein.



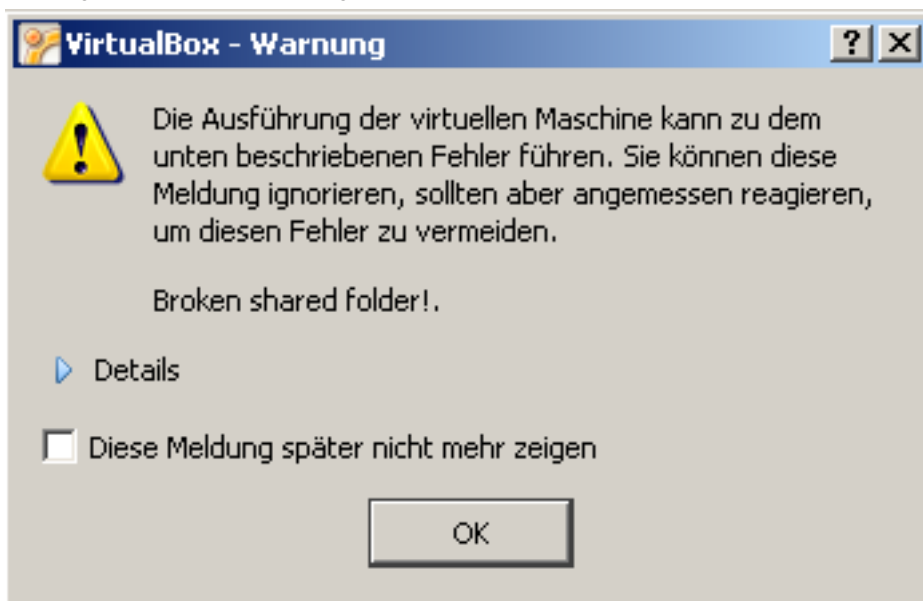
Wählen Sie die '**Automatisch einbinden**' und '**Permanent erzeugen**' an.



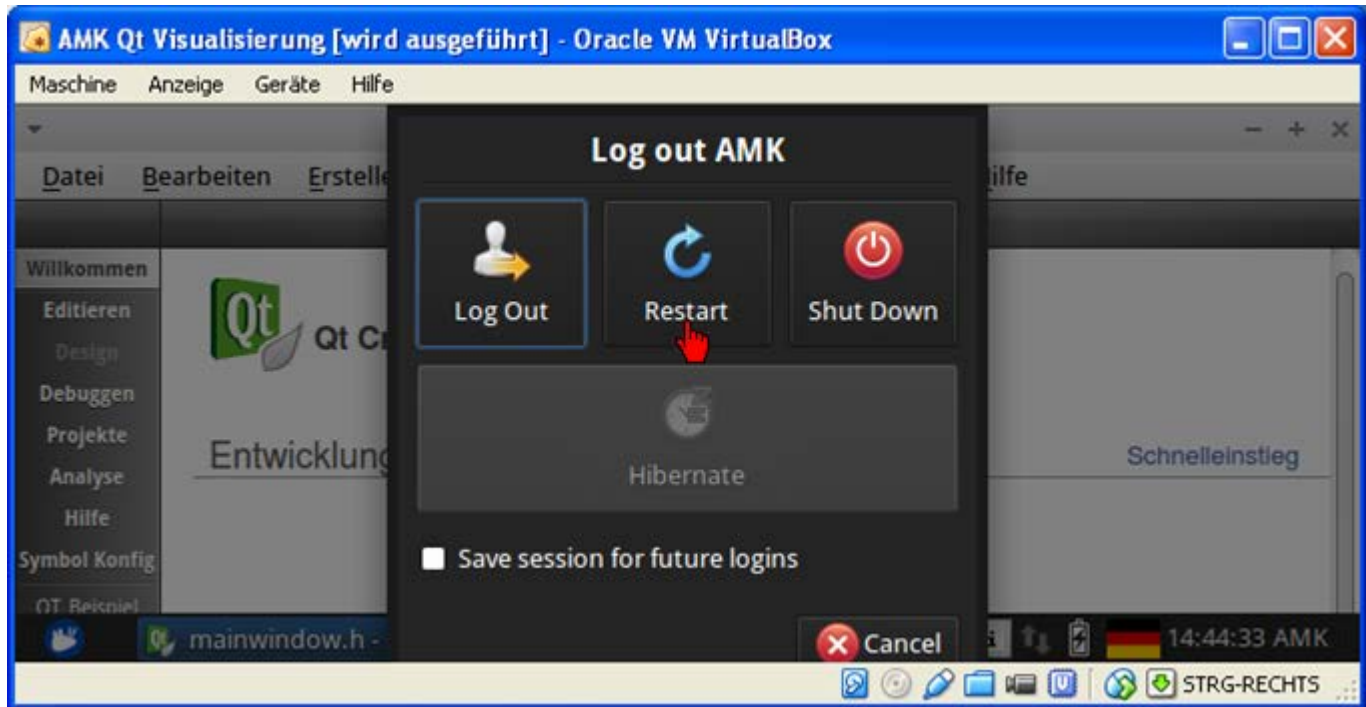
Bestätigen Sie mit **'OK'**.



Bestätigen Sie die Warnmeldungen mit **'OK'**. Anschließend müssen Sie den 'Qt Creator' neu starten.



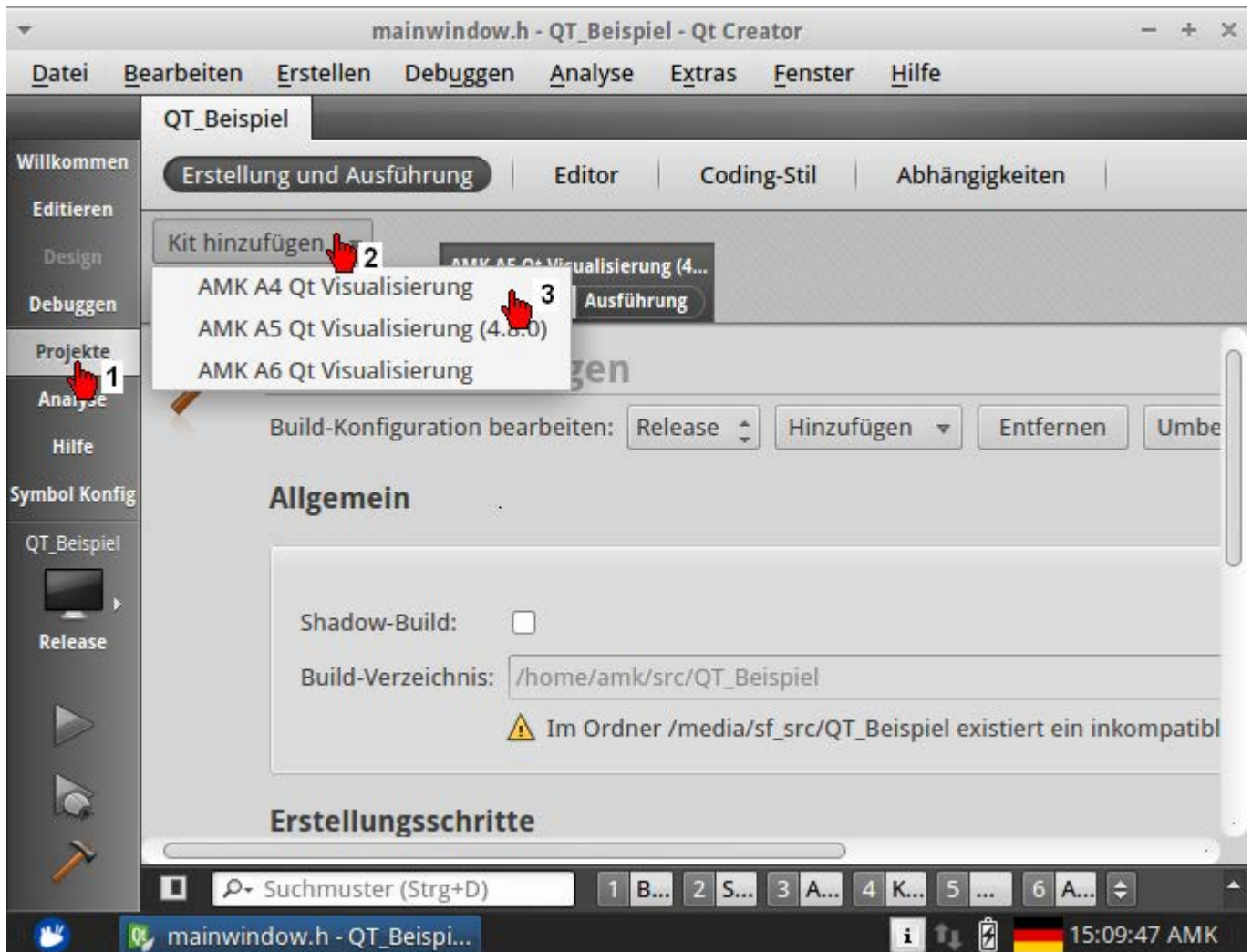
Öffnen Sie das Menü **'Log out amk'** durch klicken auf die **'Quit'** Taste rechts neben der Uhrzeit. Führen Sie anschließend einen **'Restart'** durch.



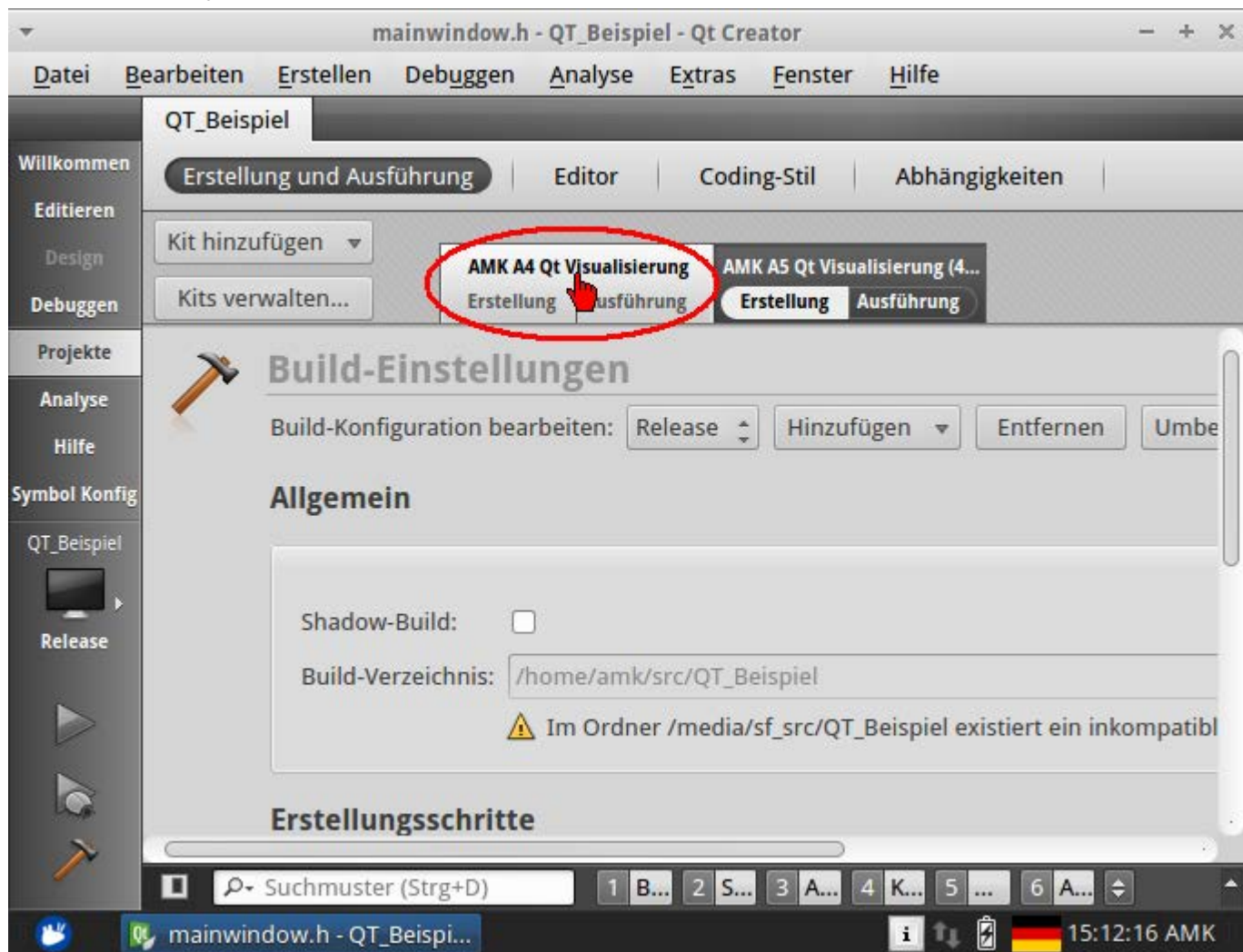
5.8.2 Umstellen eines vorhandenen Projektes auf ein anderes Zielsystem

Integrieren Sie die benötigten Zielsysteme in Ihrem Projekt.

Klicken Sie auf den Taster 'Kit hinzufügen', anschließend können Sie Ihre Auswahl treffen.



Wählen Sie das Zielsystem an, für dass das aktuelle Projekt übersetzt werden soll.
Führen Sie anschließend unter dem Menüpunkt **'Erstellen' -> 'qmake ausführen'** durch.
Erstellen Sie das Projekt neu.



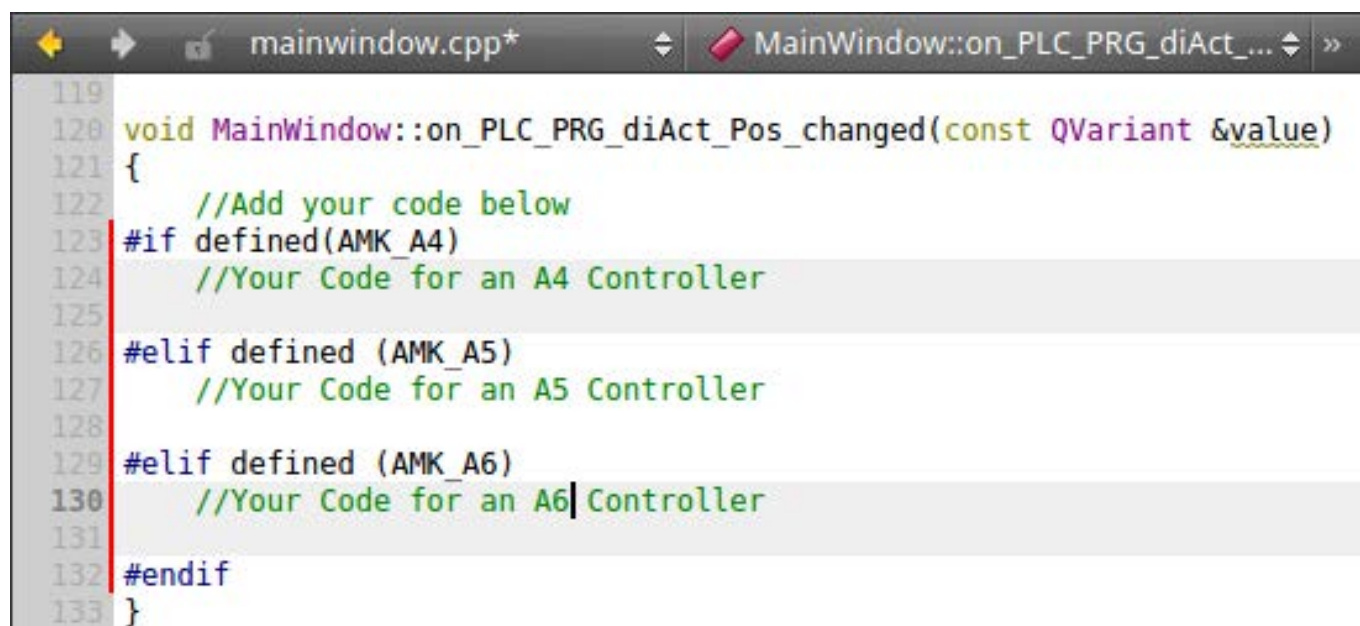
Für die Steuerung A5 sind zwei Targets enthalten. Das standardmäßig ausgewählte Target verwendet die Toolchain mit der Qt Version 4.7.0. Das zweite Target verwendet dagegen die Toolchain mit der Qt Version 4.8.0.



Für Steuerungen mit der Firmware <Ax_410_0000_000000 muss immer das Toolchain 4.7.0 verwendet werden. Andere Versionen sind nicht kompatibel.

5.8.3 Unterscheidung der Steuerungstypen

In der Qt Visualisierung ab der Version 2.0 stehen insgesamt drei Defines zur Verfügung, mit deren Hilfe die unterschiedlichen Steuerungstypen unterschieden werden können.



```
119
120 void MainWindow::on_PLG_PRG_diAct_Pos_changed(const QVariant &value)
121 {
122     //Add your code below
123     #if defined(AMK_A4)
124         //Your Code for an A4 Controller
125     #elif defined (AMK_A5)
126         //Your Code for an A5 Controller
127     #elif defined (AMK_A6)
128         //Your Code for an A6 Controller
129     #endif
130 }
131
132
133
```

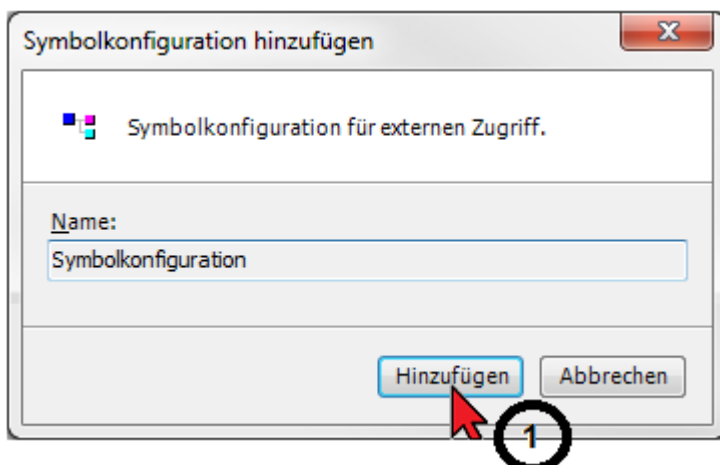
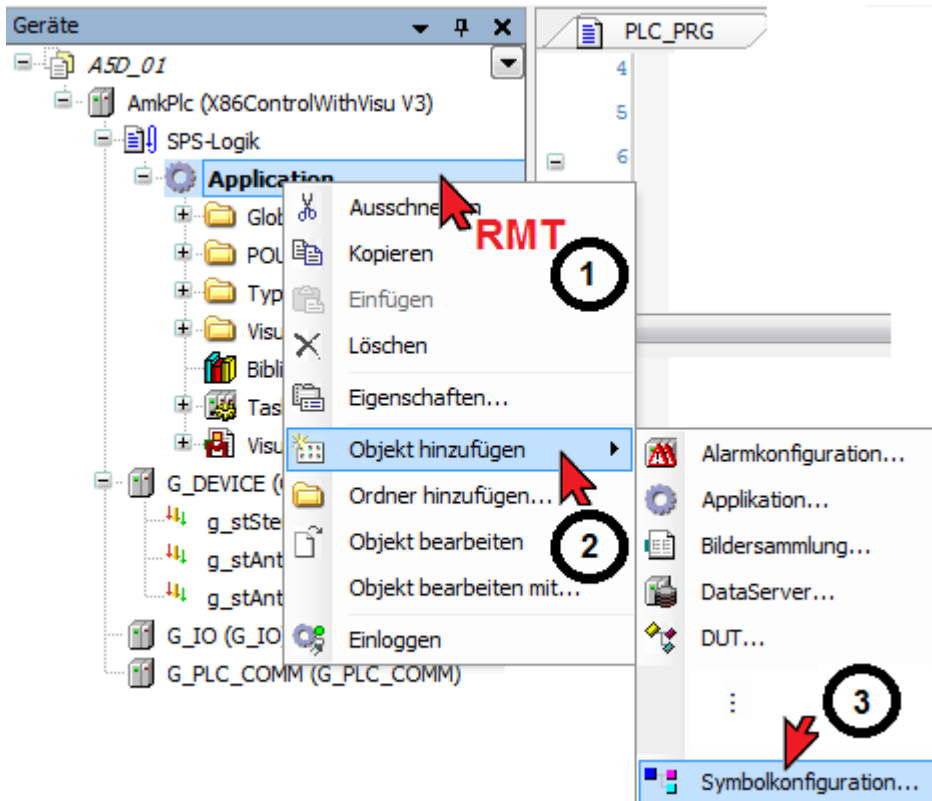
6 First Steps CODESYS V3

Das Handling der Qt Software ist im wesentlichen identisch wie unter CODESYS V2. [Siehe 'Beschreibung First Steps' auf Seite 32.](#)

Das Handling der CODESYS V3 Software ist in diesem Kapitel beschrieben.

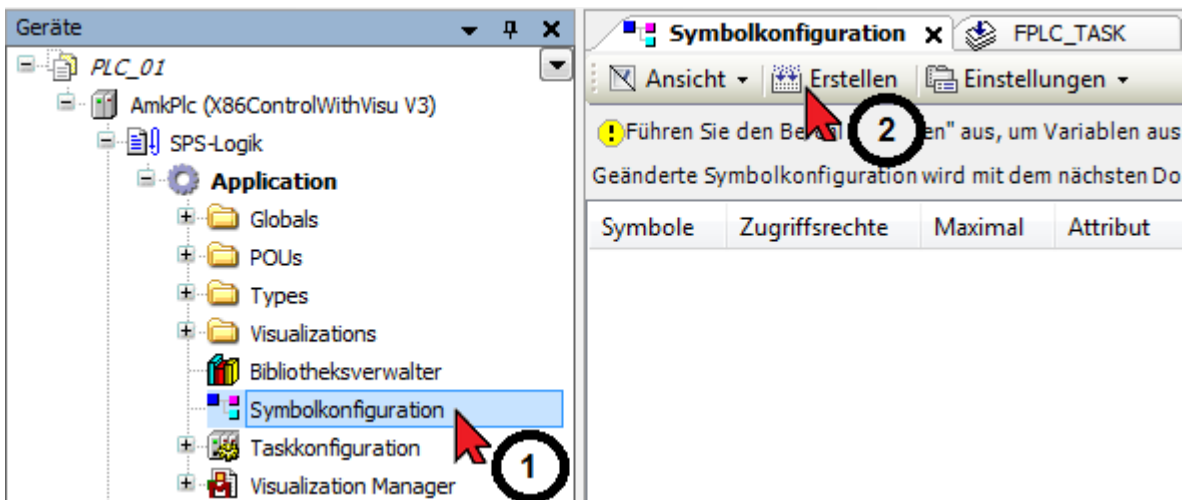
6.1 Symbolkonfiguration hinzufügen

Die Symbolkonfiguration wird verwendet, um Kommunikationsvariablen zwischen CODESYS und Qt auszutauschen.



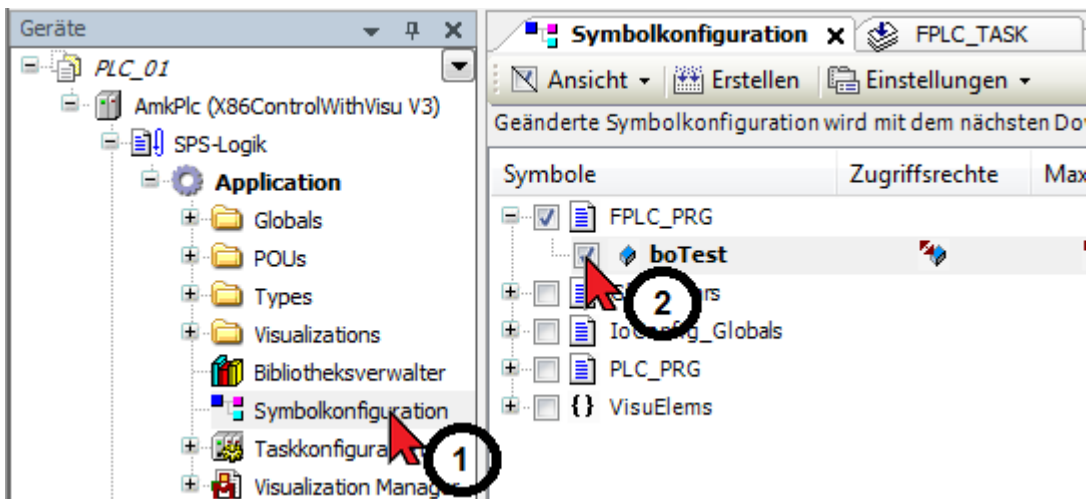


Führen Sie erst den Befehl 'Erstellen' aus, danach können Sie die Kommunikationsvariablen auswählen (ein fehlerfreier Build ist erforderlich).



Die markierten Variablen werden in die Symbolkonfiguration übernommen.

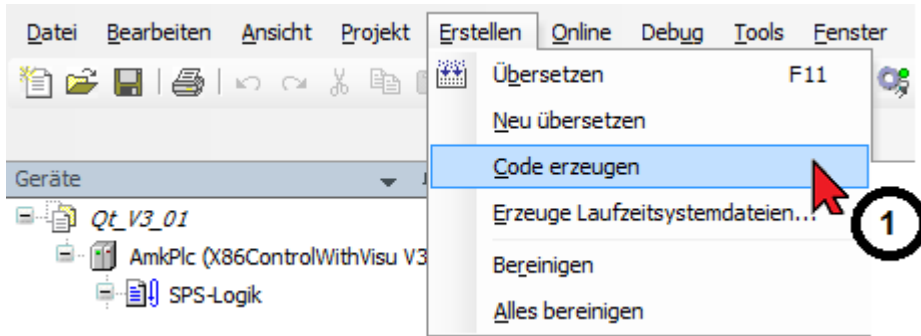
Wählen Sie die Qt Testvariable 'boTest' aus.



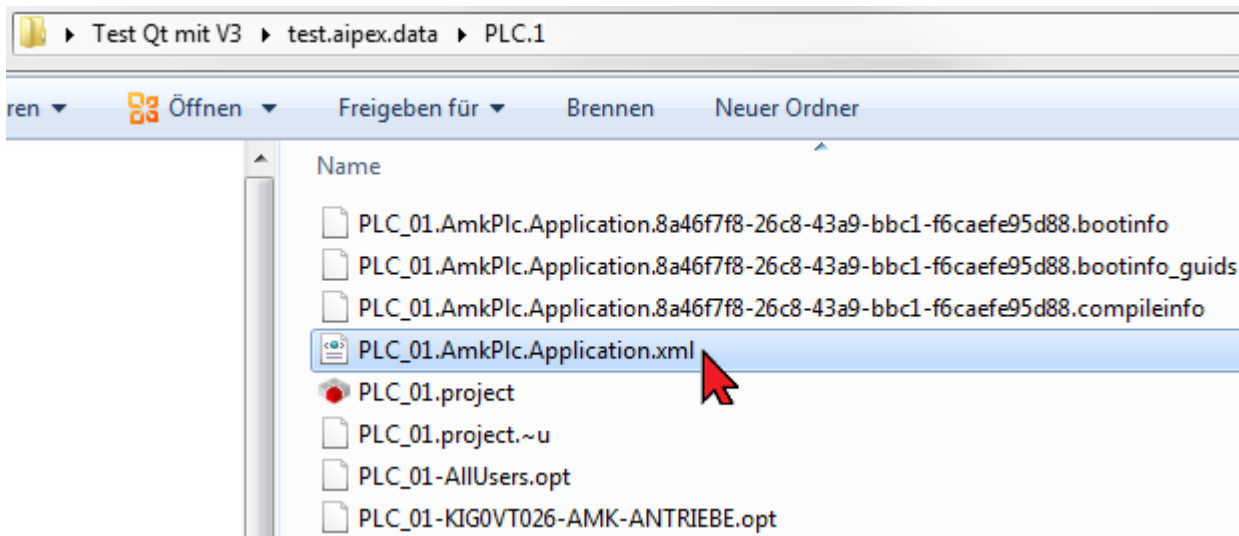


Die geänderte Symbolkonfiguration wird mit dem nächsten 'Download' oder 'Online Change' übertragen.

Folgende CODESYS Funktion Menü: 'Erstellen' → 'Code erzeugen' erstellt die Datei ohne Download.



Die Symbolkonfiguration (*.xml) wird im AIPEX PRO Projektordner gespeichert.



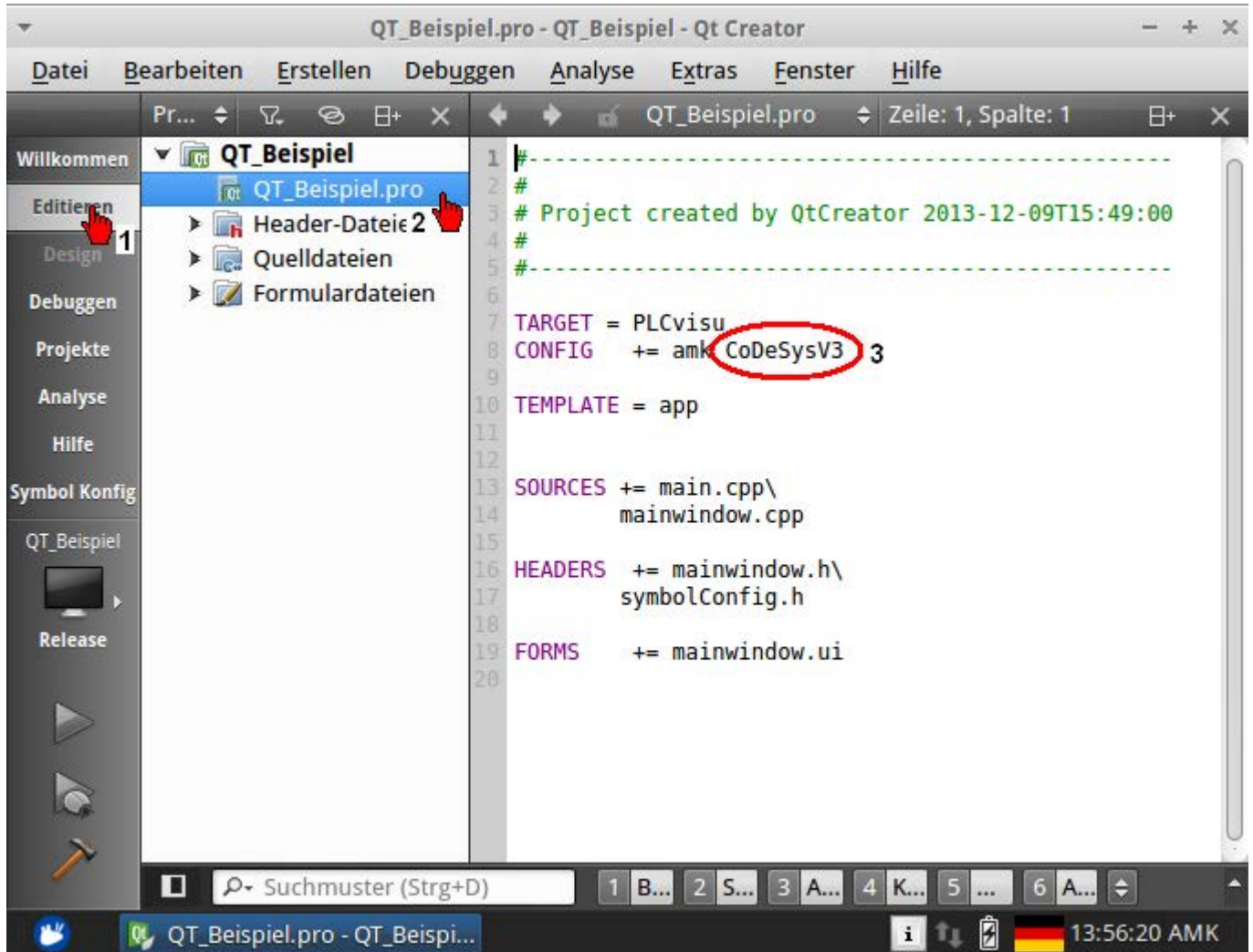
Geöffnete Symbolkonfiguration (*.xml) mit der Testvariable 'boTest'.

```
<?xml version="1.0" encoding="UTF-8"?>
- <Symbolconfiguration xmlns="http://www.3s-software.com/schemas/Symbol
  + <Header>
  + <TypeList>
  - <NodeList>
    - <Node name="Application">
      - <Node name="FPLC_PRG">
        <Node name="boTest" access="ReadWrite" type="T_BOOL"/>
      </Node>
    </Node>
  </NodeList>
</Symbolconfiguration>
```

6.2 Erweiterung im Qt Quellcode

Erweitern Sie die *.pro Datei, damit die 'Qt Visualisierung' unter CODESYS V3 verwendet werden kann.

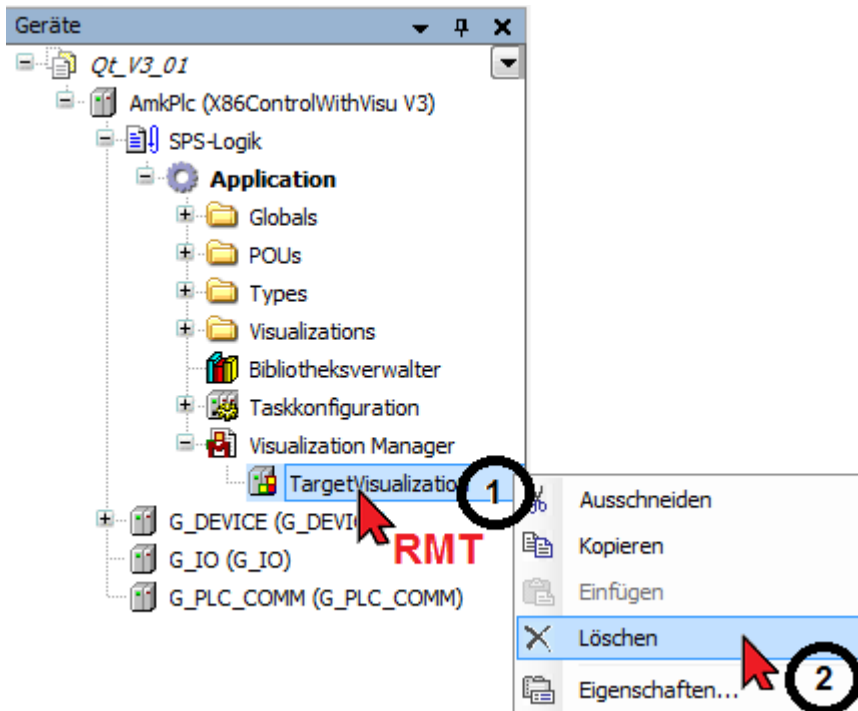
- Klicken Sie auf **'Editieren'**
- Öffnen Sie die Datei *.pro
- Ergänzen Sie in der Datei *.pro den Eintrag CONFIG mit 'CODESYSV3'
- Führen Sie unter dem Menüpunkt **'Erstellen'** den Punkt **'qmake ausführen'** durch
- Erstellen Sie das Projekt neu (z.B. unter dem Menüpunkt **'Erstellen'**)



6.3 Qt Visualisierung integrieren

Das Kapitel beschreibt, wie Sie die vom 'Qt Creator' erstellte Datei 'PLCvisu' in das CODESYS V3 Projekt integrieren.

Löschen Sie (wenn vorhanden) die Target-Visualisierung.

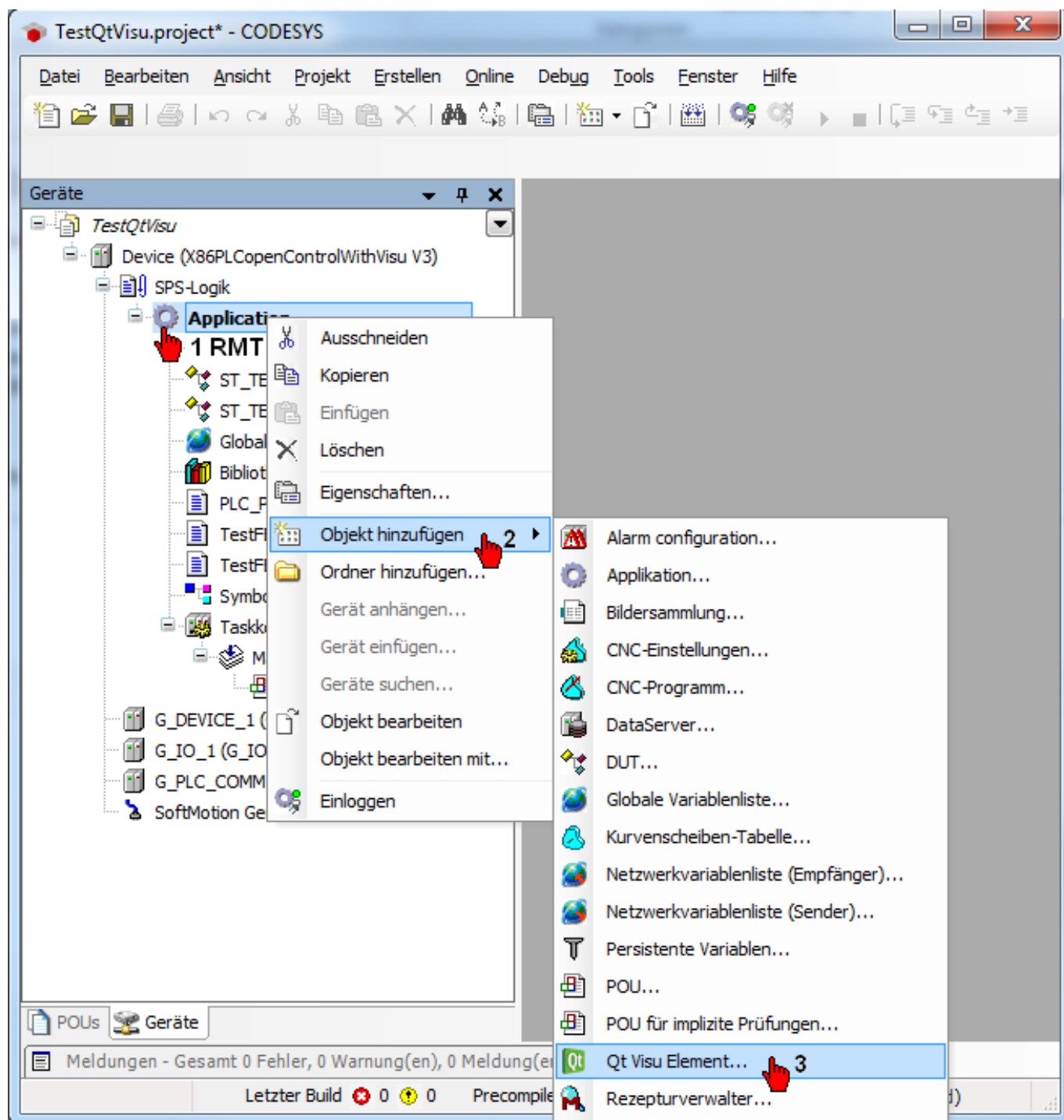


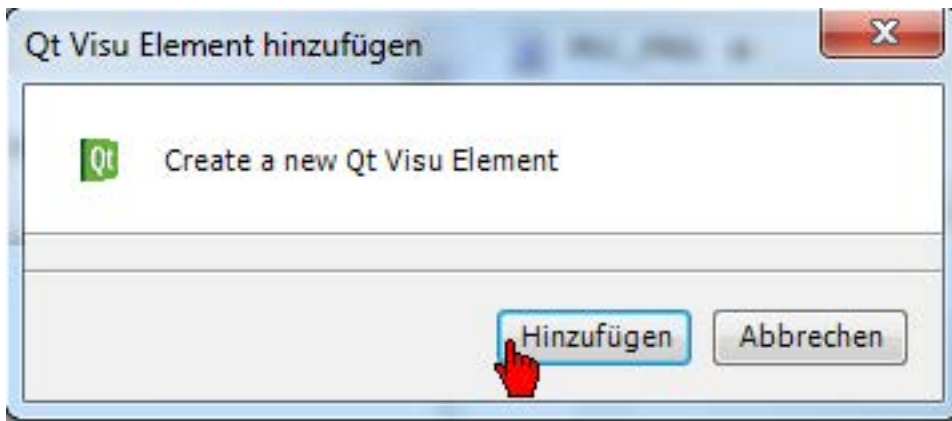
Erstellen Sie ein 'Qt Visu Element ...'



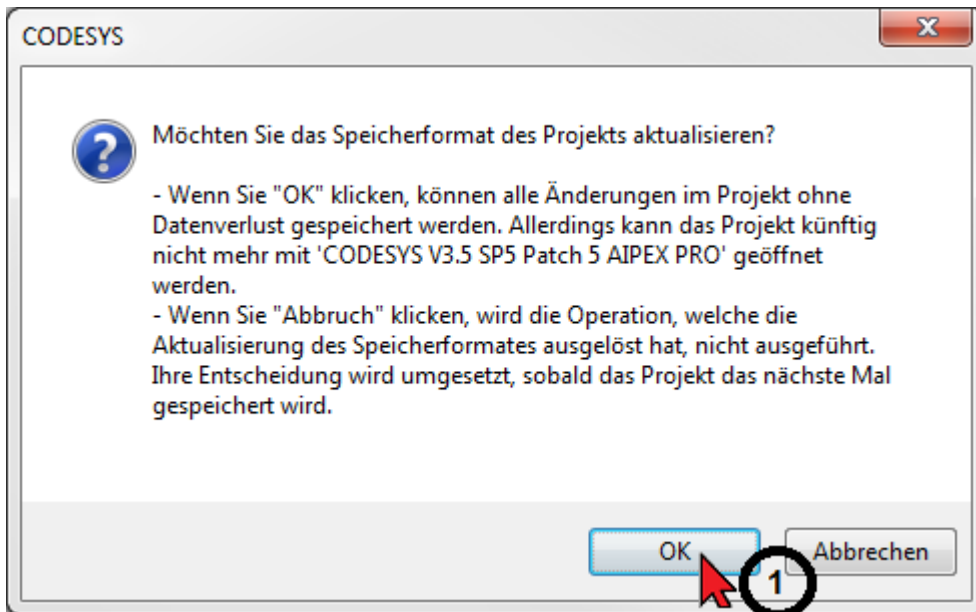
Voraussetzungen: Installiertes 'Qt AddOn'

Siehe Installation 'Qt AddOn für CODESYS V3' auf Seite 17.





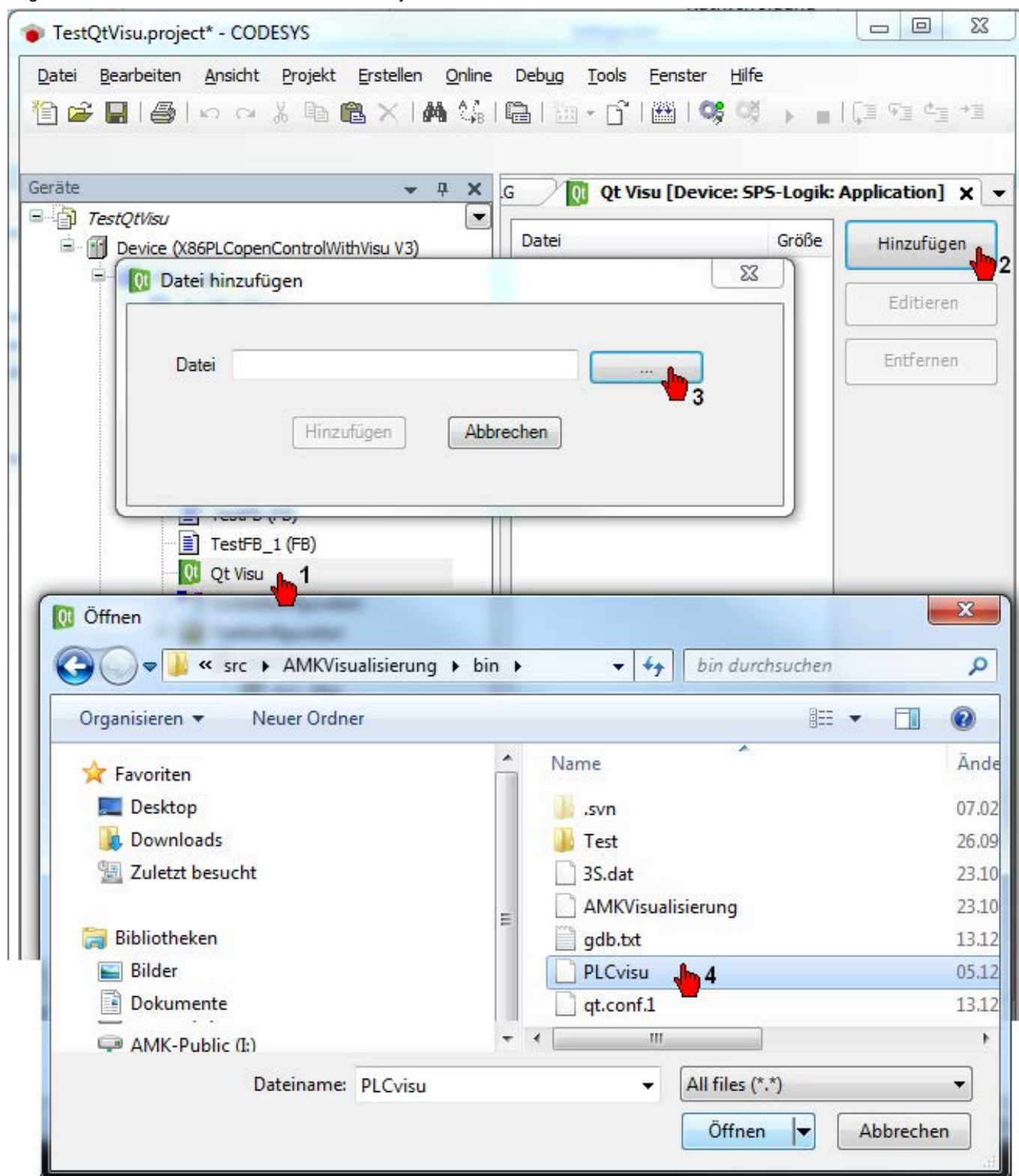
Bestätigen Sie mit 'OK'. Das Öffnen mit der angegebenen CODESYS Version ist trotzdem möglich.



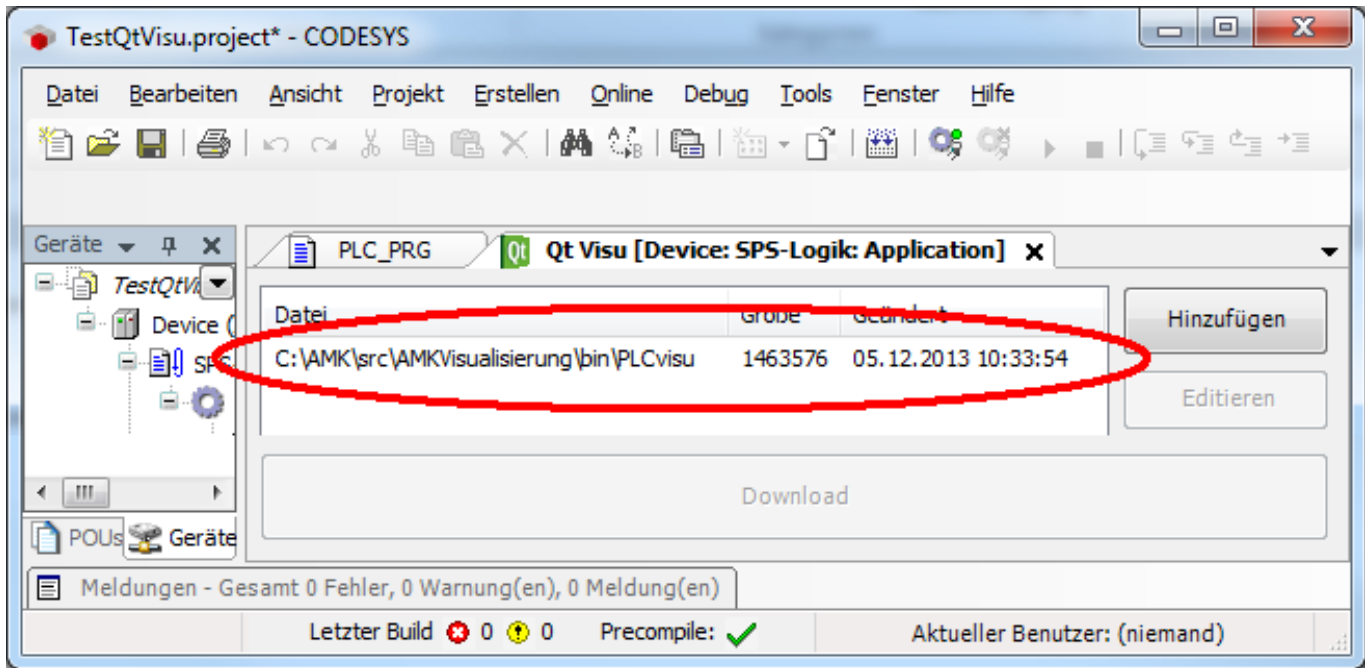
Der Qt Creator erstellte beim Übersetzen eine Datei mit dem Namen PLCvisu.bin.

Die Defaulteinstellung für den Pfad lautet c:\AMK\src\Projektname\bin\PLCvisu.

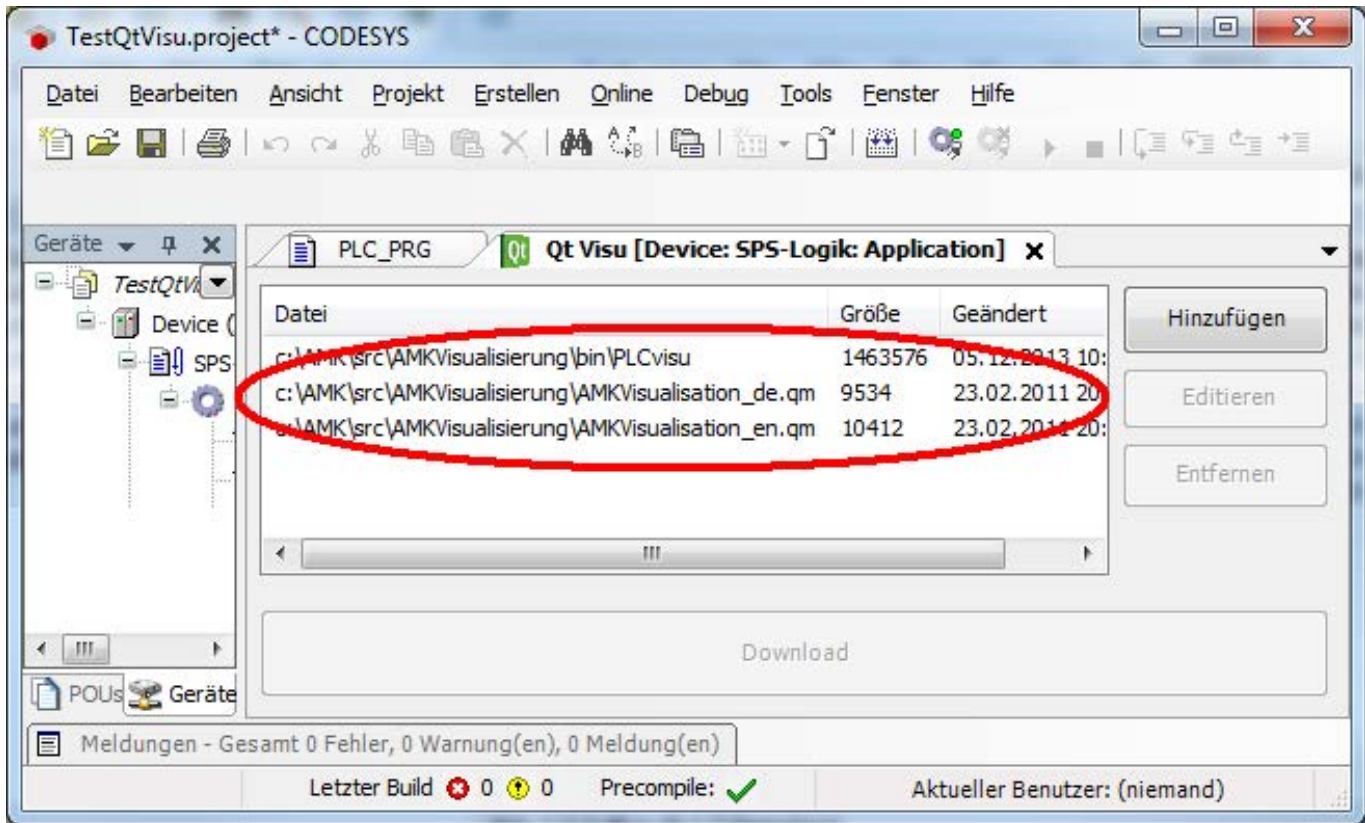
Fügen Sie die Datei PLCvisu in Ihr CODESYS Projekt ein.



CODESYS zeigt den Pfadnamen, die Dateigröße und das Änderungsdatum an.

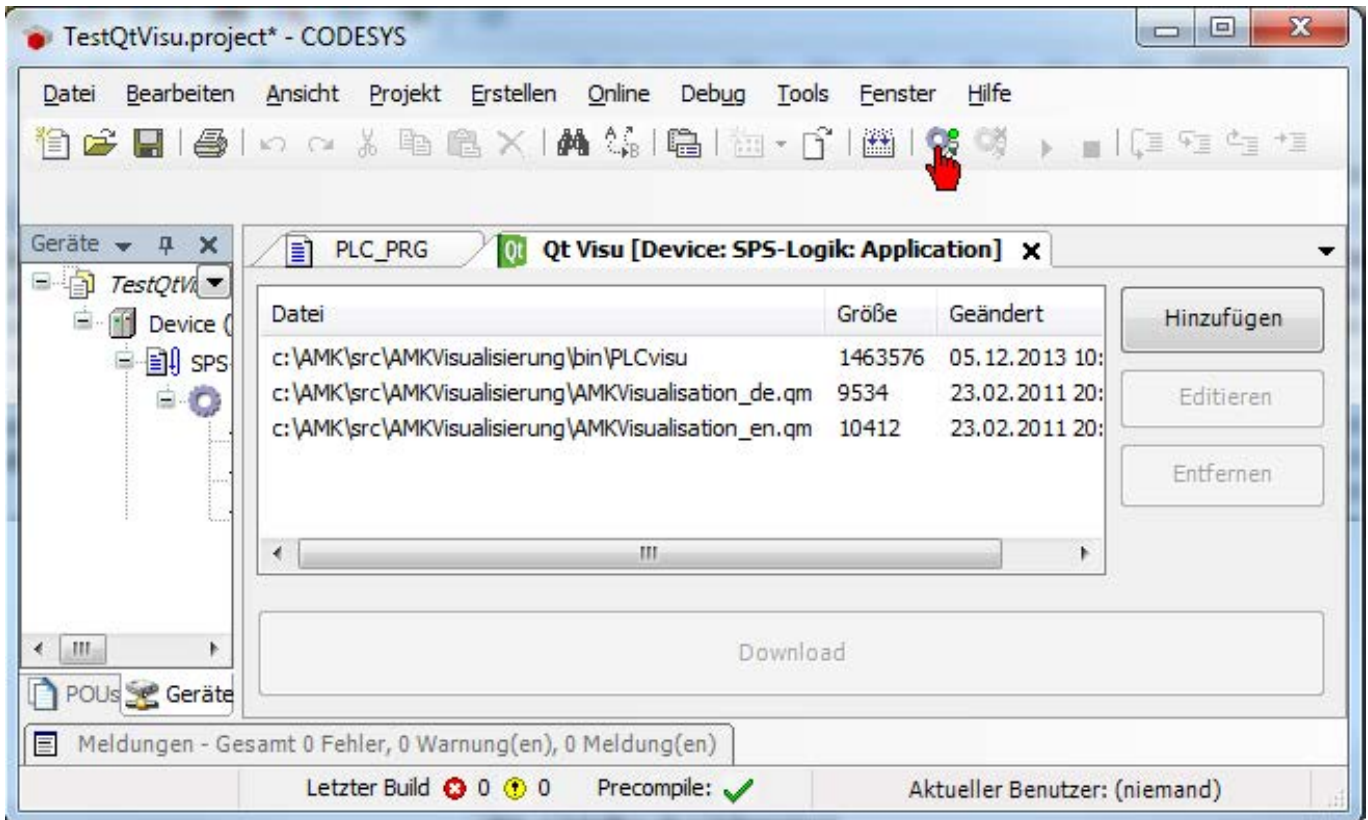


Ergänzen Sie das 'Qt Visu Element' mit allen Dateien die auf die Steuerung geladen werden sollen. Z.B Übersetzungsdateien des Programms 'Qt Linguist'.



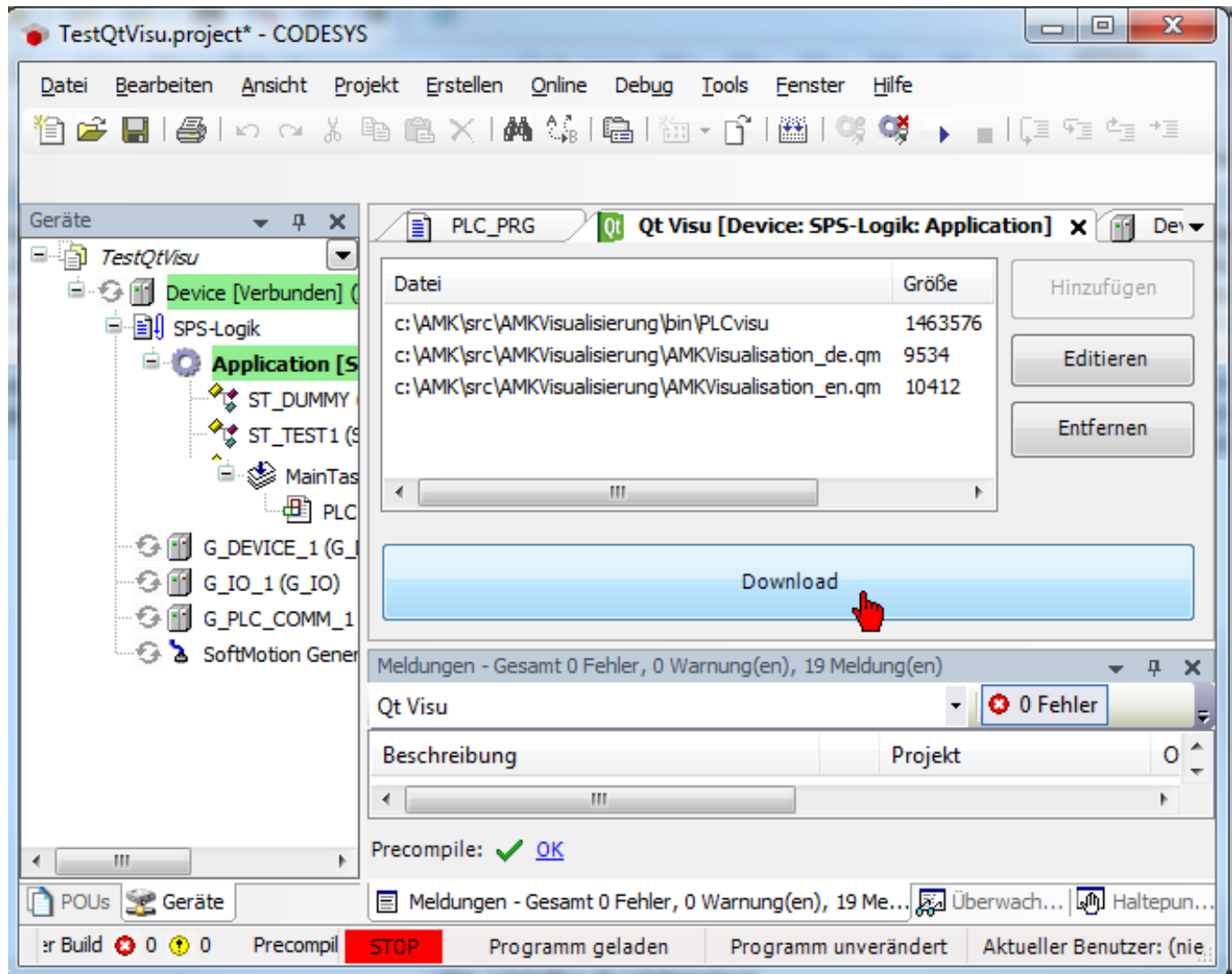
6.3.1 Projekt übertragen

Beim 'Einloggen' wird das 'Qt Visu Element' mit den Dateien aus dem 'Qt Creator' auf die Steuerung übertragen.



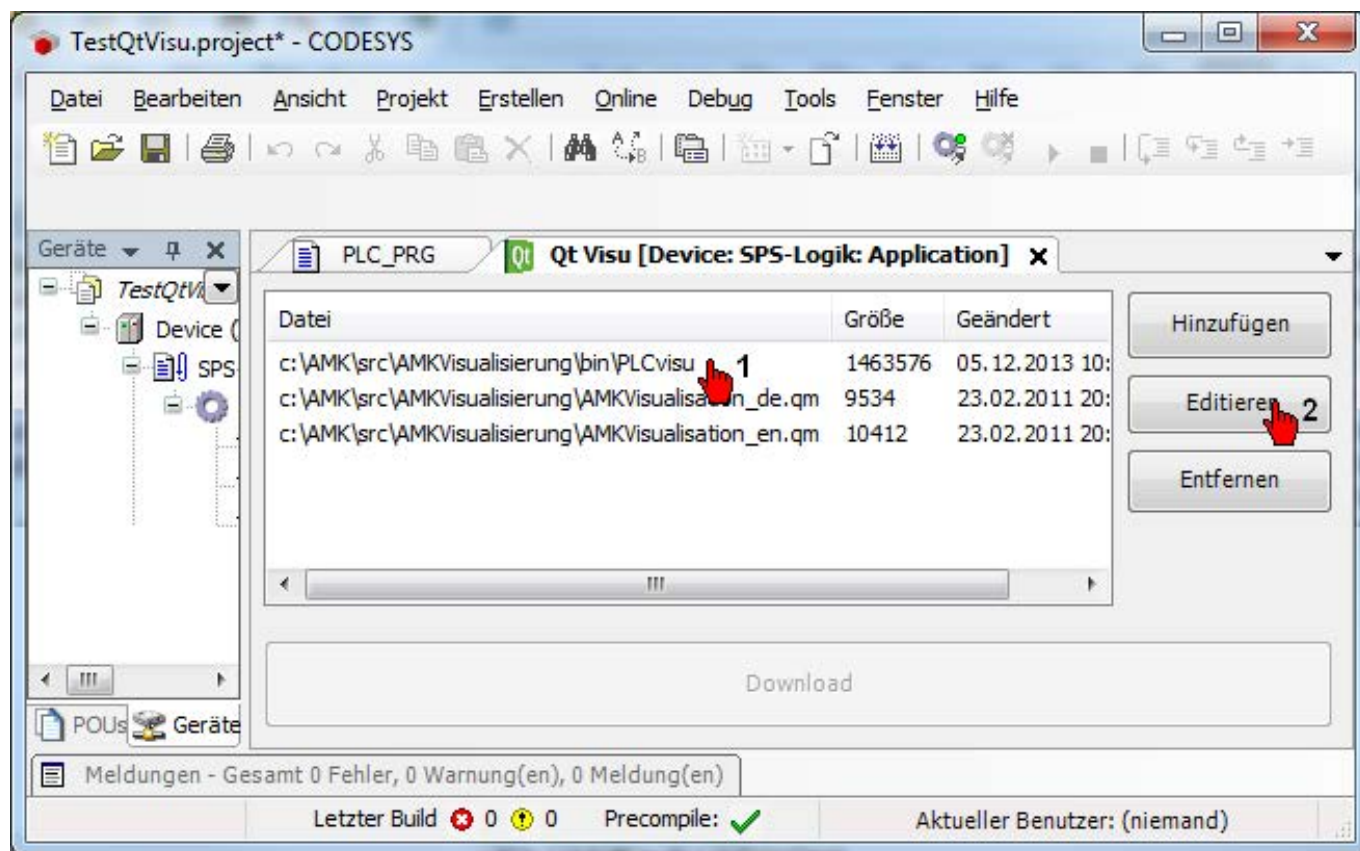
Mit dem Taster **'Download'** werden alle Qt Visu Dateien, die seit dem letzten Download aktualisiert wurden, auf die Steuerung geladen.

Voraussetzung: CODESYS ist mit der Steuerung verbunden (online).

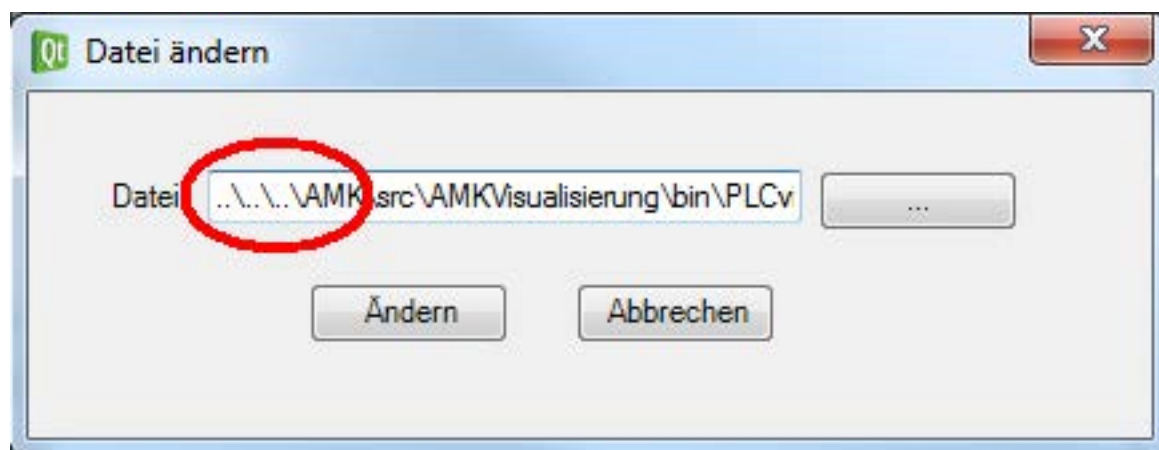


6.3.2 Pfadname absolut / relativ

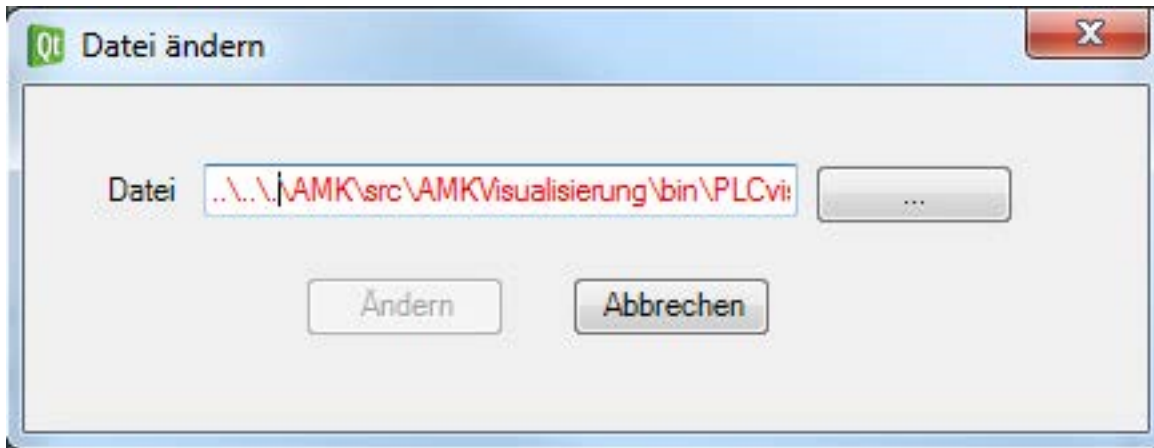
Ändern des absoluten Pfadnamens auf einen relativen Pfadnamen



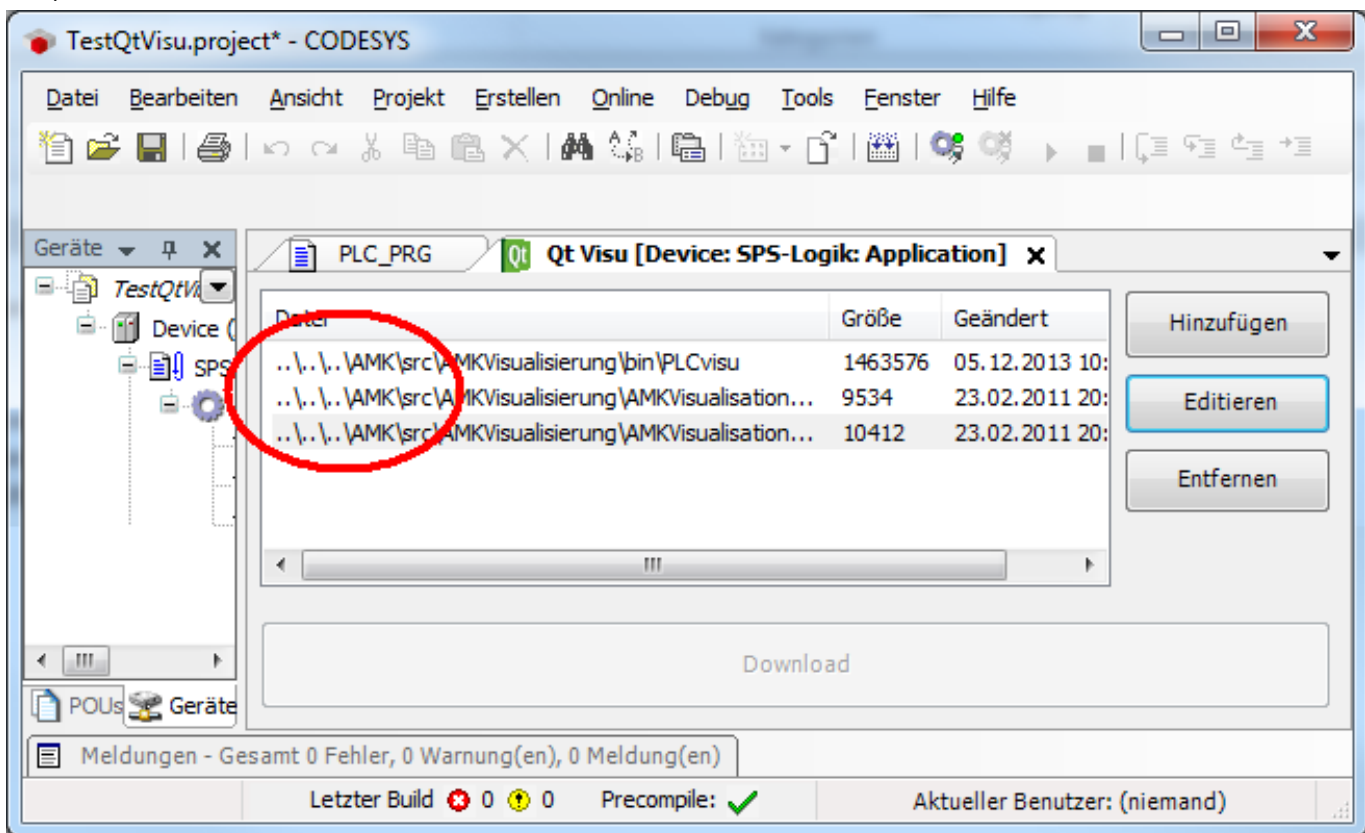
Geben Sie den relativen Pfadnamen ein



Roter Pfad: Ungültige Pfadangabe bzw. Datei nicht vorhanden



Beispiel:



7 Programmierung QtPlcHandler

Der QtPlcHandler ist eine C++-Klasse, die auf einer komfortablen Ebene Dienste für die Kommunikation einer Visualisierung zu einer AMK Steuerung anbietet.

Folgende Features und Dienste sind enthalten:

- Verbindungsauf- und Abbau zu einer Steuerung
- Lesen aller in der Steuerung verfügbaren Variablen
- Zyklisches Lesen von Variablenwerten aus der Steuerung
- Synchrones Lesen von Variablenwerten aus der Steuerung
- Synchrones Schreiben von Variablenwerten in die Steuerung
- Upload / Download von Dateien (Remote System)
- Dateien löschen / umbenennen
- Verzeichnisstruktur auslesen
- Projektinformationen lesen
- SPS Starten / Stoppen
- Bootprojekt Neustarten
- Status QtPlcHandler / SPS auslesen

7.1 Konfiguration und Verbindung

Konstruktor und Konfiguration des PlcHandlers

Ein QtPlcHandler Objekt ist immer einer einzelnen Steuerung zugeordnet. Beim Anlegen des Objektes muss daher konfiguriert werden, mit welcher Steuerung der QtPlcHandler arbeiten soll.

Es gibt zwei verschiedene Konstruktoren zum Erzeugen des QtPlcHandler Objektes:

- 1 explicit QtPlcHandler(QObject *parent=0);

Dieser Konstruktor muss verwendet werden, wenn die Visualisierung lokal auf dem Gerät läuft.

Parameter:

parent: optional für Objekthierarchie von Qt

- 2 explicit QtPlcHandler(const QString &ip, QObject *parent=0);

Dieser Konstruktor muss verwendet werden, wenn die Visualisierung nicht lokal auf dem Gerät läuft und die Verbindung zur Steuerung über Netzwerk realisiert wird.

Hier muss als erster Parameter die IP-Adresse als String übergeben werden (z.B. „172.16.4.101“)

Parameter:

Ip: IP Adresse als String z.B. „172.16.4.101“)

parent: optional für Objekthierarchie von Qt

Destruktor (Verbindungsabbau)

Wenn das QtPlcHandler Objekt gelöscht wird, wird die Verbindung zur Steuerung getrennt und alle Ressourcen freigegeben sowie alle internen Threads beendet.

Verbindungsaufbau

Methode: bool ::connectPlc();

Mit dieser Methode wird die Verbindung zur Steuerung aufgebaut. Als Ergebnis liefert die Methode TRUE für einen erfolgreichen Verbindungsaufbau und FALSE, wenn keine Verbindung zur Steuerung aufgebaut werden konnte.

Verbindungsabbau

Methode: bool ::disconnectPlc();

Mit dieser Methode wird die Verbindung zur Steuerung abgebaut. Als Ergebnis liefert die Methode TRUE für einen erfolgreichen Verbindungsabbau und FALSE, wenn die Verbindung zur Steuerung nicht abgebaut werden konnte.



Wenn ein Fehler beim Verbindungsaufbau oder -abbau auftritt, kann mit der Methode 'QString getLastError()' der zuletzt aufgetretene Fehler ausgelesen werden.

7.2 Status des QtPlcHandlers

Methode: `long ::getState() const;`

Mit dieser Methode kann der aktuelle Status des QtPlcHandlers abgefragt werden. Als Ergebnis liefert die Methode den aktuellen Status zurück.

Status

Define	Wert	Beschreibung
STATE_TERMINAT	-1	QtPlcHandler baut sich gerade ab (Destruktor)
STATE_PLC_NOT_CONNECTED	0	Steuerung nicht verbunden (Anfangs-Status)
STATE_PLC_CONNECTED	1	Steuerung ist verbunden
STATE_NO_SYMBOLS	2	Steuerung ist verbunden, aber es sind noch keine Symbole geladen
STATE_SYMBOLS_LOADED	3	Steuerung ist verbunden und die Symbole sind geladen
STATE_RUNNING	4	Steuerung ist verbunden, die Symbole sind geladen und alle verifiziert. Mit dem QtPlcHandler Objekt kann korrekt gearbeitet werden.
STATE_DISCONNECT	5	Verbindung zur Steuerung wird gerade abgebaut
STATE_NO_CONFIGURATION	6	Es ist keine gültige Konfiguration vorhanden
STATE_PLC_NOT_CONNECTED_SYMBOLS_LOADED	7	Symbole sind Offline verfügbar

7.3 Symbol-Informationen

Methode: `QList <PlcSymbolDesc> * ::getSymbols();`

Mit dieser Methode können alle Symbol-Informationen abgerufen werden. Als Ergebnis wird ein Pointer auf eine Liste mit allen Symbol-Informationen zurück gegeben.

Aufbau der Struktur 'PlcSymbolDesc':

struct PlcSymbolDesc	
{	
char* pszName;	symbolischer Name der Variablen
unsigned long ulTypeld;	Referenz auf Symbol-Typtabelle
char* pszType;	Typname der Variablen
unsigned short usRefld;	Variablenart (input, output, global, ...)
unsigned long ulOffset;	Offset im Speicherabbild
unsigned long ulSize;	Variablengröße
char szAccess[2];	Zugriffstyp der Variablen:
	no kein Zugriff
	read Lesen
	write Schreiben
	both Lesen und Schreiben
unsigned char bySwapSize;	Swap-Größe der Variablen
};	

**Besonderheit:**

Im CODESYS Laufzeitsystem ist die Größe eines Bit Symbols per Definition = 0. Deshalb muss die Symbolgröße auf 0 überprüft werden und eine spezielle Bit-Codierung vorgenommen werden (z.B. memset() oder memcpy() mit Parameter size=1 anstelle der ursprünglichen Symbolgröße).

Methode:

QList <QString> *::getSymbolNames();

Mit dieser Methode können alle Symbol-Namen abgerufen werden. Als Ergebnis wird ein Pointer auf eine Liste mit allen Symbol-Namen zurück gegen.

7.4 Zyklisches Update von Variablen



Wenn bei einer der Methoden ein Fehler auftritt, kann mit der Methode 'QString getLastError()' der zuletzt aufgetretene Fehler ausgelesen werden.

Anlegen einer zyklischen Liste**Methode:**

H CYCLIST ::setCyclicList(const QStringList &plcVariableList, unsigned long updateRate=100);

Mit dieser Methode kann eine Liste von Variablen angemeldet werden, deren Werte zyklisch vom QtPlcHandler aktualisiert und jederzeit ausgelesen werden können. Als Rückgabewert wird ein Handle auf eine zyklische Liste übergeben. Dieses Handle muss beim Löschen der Liste mit übergeben werden. Ist der Rückgabewert NULL, konnte die zyklische Liste nicht angelegt werden.

Parameter:

plcVariableList	Stringliste mit allen Variablennamen, die zyklisch aktualisiert werden sollen
updateRate	Aktualisierungsrate [ms], Default: 100 ms

Freigeben einer zyklischen Liste**Methode:**

bool ::deleteCycVarList(const H CYCLIST handle);

Mit dieser Methode wird die zuvor angelegte zyklische Liste gelöscht. Dazu muss das beim Anlegen der Liste zurückgegebene Handle übergeben werden. Als Ergebnis liefert die Methode TRUE für einen erfolgreichen Löschen der zyklischen Liste und FALSE, wenn die zyklische Liste nicht gelöscht werden konnte.

Parameter:

handle	Handle der Liste, die gelöscht werden soll
--------	--

Aktualisierungsrate einstellen

Methode: `bool ::setUpdateRate(const HCYCLIST handle, const unsigned long updateRate);`
 Mit dieser Methode kann die Aktualisierungsrate der zyklischen Liste verändert werden. Als Ergebnis liefert die Methode TRUE für einen erfolgreichen Verändern der Aktualisierungsrate und FALSE, wenn die Aktualisierungsrate nicht verändert werden konnte.

Parameter:
 handle Handle der Liste, deren Aktualisierungsrate geändert werden soll
 updateRate Aktualisierungsrate [ms]

Aktuelle Aktualisierungsrate abfragen

Methode: `unsigned long ::getCyclicOperatingRate(const HCYCLIST handle);`
 Mit dieser Methode kann die aktuell eingestellte Aktualisierungsrate der entsprechenden zyklischen Liste abgefragt werden. Als Rückgabe erfolgt die aktuelle Aktualisierungsrate.

Parameter:
 handle Handle der Liste, deren Aktualisierungsrate abgefragt werden soll

7.5 SPS Programm steuern

SPS Programm starten

Methode: `bool ::startPlc();`
 Mit dieser Methode kann das SPS Programm gestartet werden. Als Ergebnis liefert die Methode TRUE für einen erfolgreichen Start des SPS Programms und FALSE, wenn das SPS Programm nicht gestartet werden konnte.



Diese Funktion wird in Verbindung mit einer CODESYS V3 Steuerung aktuell noch nicht unterstützt.

SPS Programm stoppen

Methode: `bool ::stopPlc();`
 Mit dieser Methode kann das SPS Programm gestoppt werden. Als Ergebnis liefert die Methode TRUE für ein erfolgreiches Stoppen des SPS Programms und FALSE, wenn das SPS Programm nicht gestoppt werden konnte.

SPS Programm Status

Methode: `QT_PLC_STATUS ::getPlcState() const;`
 Mit dieser Methode kann der Status des SPS Programms abgefragt werden. Als Rückgabewert wird der aktuelle Status zurück gegeben.

Status

Define	Wert	Beschreibung
QT_PLC_STATE_RUNNNING	0	SPS läuft
QT_PLC_STATE_STOP	1	SPS ist gestoppt
QT_PLC_STATE_STOP_ON_BP	2	SPS hat an einem Haltepunkt gestoppt
QT_PLC_STATE_UNKNOWN	255	Undefinierter Zustand

Bootprojekt neu laden

Methode: `bool ::reloadBootProject();`
 Mit dieser Methode wird das Bootprojekt neu geladen.



Achtung:
 Hierbei wird auch das Visualisierungsprogramm neu gestartet!

7.6 Manuelles Lesen von Variablen

Mit den nachfolgenden Methoden kann der Wert einer Variablen ausgelesen werden. Als Ergebnis liefern die Methoden jeweils TRUE für ein erfolgreiches Lesen und FALSE, wenn die Variable nicht gelesen werden konnte.



Wenn bei einer der Methoden ein Fehler auftritt, kann mit der Methode 'QString getLastError()' der zuletzt aufgetretene Fehler ausgelesen werden.



Wird mit dem PLC Handler ein Short Integer (SINT) aus der PLC gelesen, ist der ausgelesene Wert abhängig vom Zielsystem.

Auf z.B. einer A5 Steuerung wird der korrekte Wert -128 angezeigt.

Auf einer A4 Steuerung wird +127 ausgegeben. Deshalb muss das Ergebnis auf einer A4 Steuerung in ein signed char gecastet werden.

```
void MainWindow::getActTemperatur()
{
    IEC_SINT temperatur;
    m_plcHandler.readPlcValue(PLC_siActTemperatur,temperatur);
    //Wenn "PLC_siActTemperatur" eine negative Zahl enthält, dann wird beim Lesen der Variable
    //eine negative Zahl in die Variable „temperatur“ geschrieben. Um den richtigen wert zu
    //erhalten, muss die Zahl gecastet werden. Dies ist nur bei einer A4 – Steuerung notwendig.
    signed char temp_korr = (signed char) temperatur;
    //temp_korr enthält nun die vorzeichenrichtige Zahl
}
```

Methode: bool ::readPlcValue(const QString &variable, IEC_BOOL &value);

Parameter:

variable Variablen-Name als String.

value Variable vom Typ IEC_BOOL, in der der Wert der Variablen gespeichert werden soll.

Methode: bool ::readPlcValue(const QString &variable, bool &value);

Parameter:

variable Variablen-Name als String.

value Variable vom Typ bool, in der der Wert der Variablen gespeichert werden soll.

Methode: bool ::readPlcValue(const QString &variable, IEC_INT &value);

Parameter:

variable Variablen-Name als String.

value Variable vom Typ IEC_INT, in der der Wert der Variablen gespeichert werden soll.

Methode: bool ::readPlcValue(const QString &variable, IEC_DINT &value);

Parameter:

variable Variablen-Name als String.

value Variable vom Typ IEC_DINT, in der der Wert der Variablen gespeichert werden soll.

Methode: bool ::readPlcValue(const QString &variable, IEC_WORD &value);

Parameter:

variable Variablen-Name als String.

value Variable vom Typ IEC_WORD, in der der Wert der Variablen gespeichert werden soll.

Methode: bool ::readPlcValue(const QString &variable, IEC_DWORD &value);

Parameter:

variable Variablen-Name als String.

value Variable vom Typ IEC_DWORD, in der der Wert der Variablen gespeichert werden soll.

Methode:	bool ::readPlcValue(const QString &variable, IEC_REAL &value);
Parameter:	
variable	Variablen-Name als String.
value	Variable vom Typ IEC_REAL, in der der Wert der Variablen gespeichert werden soll.
Methode:	bool ::readPlcValue(const QString &variable, IEC_LREAL &value);
Parameter:	
variable	Variablen-Name als String.
value	Variable vom Typ IEC_LREAL, in der der Wert der Variablen gespeichert werden soll.
Methode:	bool ::readPlcValue(const QString &variable, QString &value);
Parameter:	
variable	Variablen-Name als String.
value	Variable vom Typ QString, in der der Wert der Variablen gespeichert werden soll.
Methode:	bool ::readPlcValues(QList <PLC_VARIABLE> &plcValues);
	Mit dieser Methode werden mehrere Variablen auf einmal gelesen.
Parameter:	
plcValues	Liste vom Typ PLC_VARIABLE, in die die Werte der Variablen eingetragen werden sollen.
Aufbau der Struktur 'PLC_VARIABLE':	
struct PLC_VARIABLE	
{	
QString name;	Variablenname
QVariant value;	Variablenwert
};	

7.7 Manuelles Schreiben von Variablen

Mit den nachfolgenden Methoden kann der Wert einer Variablen geschrieben werden. Als Ergebnis liefern die Methoden jeweils TRUE für ein erfolgreiches Schreiben und FALSE, wenn die Variable nicht geschrieben werden konnte.



Wenn bei einer der Methoden ein Fehler auftritt, kann mit der Methode 'QString getLastError()' der zuletzt aufgetretene Fehler ausgelesen werden.

Methode:	bool ::writePlcValue(const QString &variable, IEC_BOOL value);
Parameter:	
variable	Variablen-Name als String
value	Variable vom Typ IEC_BOOL, aus der der Wert der Variablen gelesen werden soll
Methode:	bool ::writePlcValue(const QString &variable, bool value);
Parameter:	
variable	Variablen-Name als String
value	Variable vom Typ bool, aus der der Wert der Variablen gelesen werden soll
Methode:	bool ::writePlcValue(const QString &variable, IEC_INT value);
Parameter:	
variable	Variablen-Name als String
value	Variable vom Typ IEC_INT, aus der der Wert der Variablen gelesen werden soll
Methode:	bool ::writePlcValue(const QString &variable, IEC_DINT value);
Parameter:	
variable	Variablen-Name als String
value	Variable vom Typ IEC_DINT, aus der der Wert der Variablen gelesen werden soll
Methode:	bool ::writePlcValue(const QString &variable, IEC_WORD value);
Parameter:	
variable	Variablen-Name als String
value	Variable vom Typ IEC_WORD, aus der der Wert der Variablen gelesen werden soll

Methode:	<code>bool ::writePlcValue(const QString &variable, IEC_DWORD value);</code>
Parameter:	
variable	Variablen-Name als String
value	Variable vom Typ IEC_DWORD, aus der der Wert der Variablen gelesen werden soll
Methode:	<code>bool ::writePlcValue(const QString &variable, IEC_REAL value);</code>
Parameter:	
variable	Variablen-Name als String
value	Variable vom Typ IEC_REAL, aus der der Wert der Variablen gelesen werden soll
Methode:	<code>bool ::writePlcValue(const QString &variable, IEC_LREAL value);</code>
Parameter:	
variable	Variablen-Name als String
value	Variable vom Typ IEC_LREAL, aus der der Wert der Variablen gelesen werden soll
Methode:	<code>bool ::writePlcValue(const QString &variable, QString value);</code>
Parameter:	
variable	Variablen-Name als String
value	Variable vom Typ QString, aus der der Wert der Variablen gelesen werden soll
Methode:	<code>bool ::writePlcValues(QList <PLC_VARIABLE> plcValues);</code> Mit dieser Methode können mehrere Variablen auf einmal geschrieben werden.
Parameter:	
plcValues	Liste vom Typ PLC_VARIABLE
Aufbau der Struktur 'PLC_VARIABLE':	
	<code>struct PLC_VARIABLE</code>
	<code>{</code>
	QString name; Variablenname
	QVariant value; Variablenwert
	<code>};</code>

7.8 Datei Operationen



Die nachfolgenden Methoden sind nur dann sinnvoll, wenn das Visualisierungsprogramm auf einem Remote PC läuft.

Wenn das Visualisierungsprogramm lokal läuft, hat man Vollzugriff auf das lokale Dateisystem.



Wenn bei einer der Methoden ein Fehler auftritt, kann mit der Methode 'QString getLastError()' der zuletzt aufgetretene Fehler ausgelesen werden.

Datei Upload

Methode:	<code>bool ::uploadFile(const QString &srcfileName, const QString &dstFileName);</code>
	Mit dieser Methode kann eine Datei von der Steuerung zu einem Client übertragen werden. Als Ergebnis liefert die Methode TRUE für einen erfolgreichen Upload der Datei und FALSE, wenn die Datei nicht übertragen werden konnte.
Parameter:	
srcfileName	Dateiname/Pfad der Datei, die kopiert werden soll
dstFileName	Dateiname/Pfad, wohin die Datei kopiert werden soll.

Datei Download

Methode:	<code>bool ::downloadFile(const QString &srcfileName, const QString &dstFileName);</code>
	Mit dieser Methode kann eine Datei von einem Client zur Steuerung übertragen werden. Als Ergebnis liefert die Methode TRUE für einen erfolgreichen Download der Datei und FALSE, wenn die Datei nicht übertragen werden konnte.
Parameter:	
srcfileName	Dateiname/Pfad der Datei, die kopiert werden soll
dstFileName	Dateiname/Pfad, wohin die Datei kopiert werden soll.

Datei umbenennen

Methode: `bool ::renameFile(const QString &oldFileName, const QString &newFileName);`

Mit dieser Methode kann eine Datei umbenannt werden. Als Ergebnis liefert die Methode TRUE für einen erfolgreichen Umbenennen der Datei und FALSE, wenn die Datei nicht umbenannt werden konnte.

Parameter:

<code>oldFileName</code>	Alter Dateiname
<code>newFileName</code>	Neuer Dateiname

Datei löschen

Methode: `bool ::deleteFile(const QString &fileName);`

Mit dieser Methode kann eine Datei gelöscht werden. Als Ergebnis liefert die Methode TRUE für einen erfolgreichen Umbenennen der Datei und FALSE, wenn die Datei nicht gelöscht werden konnte.

Parameter:

<code>fileName</code>	Dateiname der Datei, die gelöscht werden soll
-----------------------	---

Verzeichnisstruktur auslesen

Methode: `bool ::readDirectory(const QString &baseDirectory, QList<PLC_FILE_INFO> &fileInfo);`

Mit dieser Methode kann die Verzeichnisstruktur auf der Steuerung ausgelesen werden. Als Ergebnis liefert die Methode TRUE für ein erfolgreiches Auslesen der Verzeichnisstruktur und FALSE, wenn die Verzeichnisstruktur nicht gelesen werden konnte.

Parameter:

<code>baseDirectory</code>	Basisverzeichnis, dessen Verzeichnisstruktur ausgelesen werden soll
<code>fileInfo</code>	Liste, in der die Verzeichnisinformationen zurück gegeben werden

Aufbau der Struktur 'PLC_FILE_INFO':

```
struct PLC_FILE_INFO
```

```
{
```

```
    QString fileName
```

```
    bool isDir
```

```
};
```

Dateiname

TRUE: ist Verzeichnis

FALSE: ist kein Verzeichnis

7.9 SPS Projektinformationen

SPS Projektinformationen

Methode: `bool ::getPlcProjectInfos(QT_PLC_PROJECT_INFO &plcProjectInfos);`
 Mit dieser Methode können die SPS-Projektinformationen ausgelesen werden. Als Ergebnis liefert die Methode TRUE für einen erfolgreichen Auslesen der Projektinformationen und FALSE, wenn die Projektinformationen nicht ausgelesen werden konnten.

CODESYS V2: Liefert die Projektinformationen des aktuellen Projekts.

CODESYS V3: Auf einer Steuerung mit CODESYS V3 können mehrere Applikationen innerhalb von einem Projekt vorkommen. Diese Funktion liefert die Projektinformationen des ersten gefundenen Projekts zurück. Es ist nicht garantiert, dass die zurückgelieferten Informationen denen des ersten Projekt im CODESYS – Projektbaum entsprechen.

Die Variable `ulProjectId` in der Struktur `QT_PLC_PROJECT_INFO` enthält in Verbindung mit einer CODESYS V3 – Steuerung immer eine 0.

Parameter:

`plcProjectInfos` Struktur vom Typ 'QT_PLC_PROJECT_INFO', in der die Projektinformationen gespeichert werden.

Aufbau der Struktur 'QT_PLC_PROJECT_INFO':

```
struct QT_PLC_PROJECT_INFO
{
    unsigned long ulProjectId;           Projekt-ID
    QDateTime dtTimestampUtc;          UTC Zeit in Sekunden seit 1.1.1970
    QString strProject;                 Projektdatenname
    QString strTitle;                   Bezeichnung
    QString strVersion;                 Version
    QString strAuthor;                  Author
    QString strDescription;              Beschreibung
};
```

Methode: `bool::getPlcProjectInfos(QList<QT_PLC_PROJECT_INFO> &plcProjectInfos);`
 Mit dieser Methode können die SPS-Projektinformationen ausgelesen werden. Als Ergebnis liefert die Methode TRUE für ein erfolgreiches Auslesen der Projektinformationen und FALSE, wenn die Projektinformationen nicht ausgelesen werden konnten.

CODESYS V2: Liefert die Projektinformationen des aktuellen Projektes zurück. Die Liste besitzt nur ein Element.

CODESYS V3: Auf einer Steuerung mit CODESYS V3 können mehrere Applikationen innerhalb von einem Projekt vorkommen. Diese Funktion liefert die Projektinformationen aller aktiven Applikationen zurück. Die Reihenfolgen der Projektinformationen muss nicht mit der Position der Applikationen im CODESYS Projektbaum übereinstimmen.

Die Variable `ulProjectId` in der Struktur `QT_PLC_PROJECT_INFO` enthält in Verbindung mit einer CODESYS V3 – Steuerung immer den Wert 0.

Diese Funktion wird ab der Steuerungssoftware A5_411_0000_xxxxxx unterstützt.

7.10 QtPlcHandler Signale

Signal:	errorMsg(const QString &error); Dieses Signal wird automatisch gesendet, wenn ein Fehler aufgetreten ist.
Parameter:	error Fehlermeldung als String
Signal:	state(const long &state); Dieses Signal wird automatisch gesendet, wenn sich der Status des QtPlcHandlers geändert hat.
Parameter:	state Neuer Status (Siehe Status des QtPlcHandlers auf Seite 96.)
Signal:	state(const QString &msg); Dieses Signal wird automatisch gesendet, wenn sich der Status des QtPlcHandlers geändert hat.
Parameter:	msg Neuer Status als String
Signal:	plcState(const QT_PLC_STATUS &plcStatus); Dieses Signal wird automatisch gesendet, wenn sich der Status der SPS geändert hat.
Parameter:	plcStatus Neuer SPS-Status (Siehe SPS Programm steuern auf Seite 98.)
Signal:	cyclicData(const QList <PLC_VARIABLE> &data, const QList<int> &index); Dieses Signal wird automatisch gesendet, wenn sich der Wert einer oder mehrerer Variablen geändert hat und diese in einer zyklischen Liste enthalten sind.
Parameter:	data Liste mit allen zyklischen Variablen index Liste von Indizes, die auf die geänderten Variablen in der Liste 'data' zeigen
Aufbau der Struktur 'PLC_VARIABLE':	
	struct PLC_VARIABLE
	{
	QString name; Variablenname
	QVariant value; Variablenwert
	};

8 Anhang

8.1 GNU General Public License (GPL V2)

Quelle: www.gnu.org

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

8.1.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

8.1.2 Terms and conditions for copying, distribution and modification

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does.

Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

8.2 GNU General Public License (GPL V3)

Quelle: www.gnu.org

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

8.2.1 Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

8.2.2 Terms and conditions

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work’s users, your or third parties’ legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation. If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program. Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.

> Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see

<<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>

This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it

under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<http://www.gnu.org/philosophy/why-not-lgpl.html>>.

8.3 GNU Lesser General Public License (LGPL V2.1)

Quelle: www.gnu.org

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

8.3.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

8.3.2 Terms and conditions for copying, distribution and modification

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the library's name and an idea of what it does.

Copyright (C) year name of author

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in
the library 'Frob' (a library for tweaking knobs) written
by James Random Hacker.

signature of Ty Coon, 1 April 1990

Ty Coon, President of Vice

That's all there is to it!

Ihre Meinung zählt!

Mit unseren Dokumentationen möchten wir Sie im Umgang mit den AMKmotion Produkten bestmöglich unterstützen.

Daher sind wir ständig bestrebt, unsere Dokumentationen zu optimieren.

Ihre Kommentare oder Anregungen sind für uns immer interessant.

Nehmen Sie sich kurz Zeit und beantworten Sie unsere Fragen. Bitte schicken Sie anschließend eine Kopie dieser Seite an AMKmotion zurück.



E-Mail: Documentation@amk-motion.com

oder

Fax-Nr.: +49 7021/50 05-199

Vielen Dank für Ihre Mithilfe.

Ihr AMKmotion Dokumentationsteam

1. Wie sind Sie mit der Optik unserer Dokumentationen zufrieden?

(1) sehr gut (2) gut (3) mäßig (4) kaum (5) nicht

2. Ist der Inhalt gut gegliedert?

(1) sehr gut (2) gut (3) mäßig (4) kaum (5) nicht

3. Ist der Inhalt verständlich dokumentiert?

(1) sehr gut (2) gut (3) mäßig (4) kaum (5) nicht

4. Haben Sie Themen in der Dokumentation vermisst?

(1) nein (2) ja, welche:

5. Fühlen Sie sich bei AMKmotion insgesamt gut betreut?

(1) sehr gut (2) gut (3) mäßig (4) kaum (5) nicht

AMKmotion GmbH + Co KG

Telefon: +49 7021/50 05-0, Telefax: +49 7021/50 05-199

E-Mail: info@amk-motion.com

Homepage: www.amk-motion.com